

# Administração de Sistemas Linux Serviços para Internet

Eduardo Lobo  
Wagner Vieira Léo  
Bruno Alves Fagundes  
Francisco Marcelo M. Lima



**A RNP** – Rede Nacional de Ensino e Pesquisa – é qualificada como uma Organização Social (OS), sendo ligada ao Ministério da Ciência, Tecnologia e Inovação (MCTI) e responsável pelo Programa Interministerial RNP, que conta com a participação dos ministérios da Educação (MEC), da Saúde (MS) e da Cultura (MinC). Pioneira no acesso à Internet no Brasil, a RNP planeja e mantém a rede Ipê, a rede óptica nacional acadêmica de alto desempenho. Com Pontos de Presença nas 27 unidades da federação, a rede tem mais de 800 instituições conectadas. São aproximadamente 3,5 milhões de usuários usufruindo de uma infraestrutura de redes avançadas para comunicação, computação e experimentação, que contribui para a integração entre o sistema de Ciência e Tecnologia, Educação Superior, Saúde e Cultura.



Ministério da  
**Cultura**

Ministério da  
**Saúde**

Ministério da  
**Educação**

Ministério da  
**Ciência, Tecnologia  
e Inovação**



# Administração de Sistemas Linux **Serviços para Internet**

Eduardo Lobo

Wagner Vieira Léo

Bruno Alves Fagundes

Francisco Marcelo M. Lima





# Administração de Sistemas Linux **Serviços para Internet**

Eduardo Lobo  
Wagner Vieira Léo  
Bruno Alves Fagundes  
Francisco Marcelo M. Lima

Rio de Janeiro  
Escola Superior de Redes  
2013

Copyright © 2013 – Rede Nacional de Ensino e Pesquisa – RNP  
Rua Lauro Müller, 116 sala 1103  
22290-906 Rio de Janeiro, RJ

Diretor Geral  
**Nelson Simões**

Diretor de Serviços e Soluções  
**José Luiz Ribeiro Filho**

## **Escola Superior de Redes**

Coordenação  
**Luiz Coelho**

Edição  
**Pedro Sangirardi**

Coordenação Acadêmica de Administração de Sistemas  
**Sergio Ricardo Alves de Souza**

Equipe ESR (em ordem alfabética)

**Celia Maciel, Cristiane Oliveira, Derlinéa Miranda, Edson Kowask, Elimária Barbosa, Lourdes Soncin, Luciana Batista, Luiz Carlos Lobato, Renato Duarte e Yve Abel Marcial.**

Capa, projeto visual e diagramação  
**Tecnodesign**

Versão  
**2.0.0**

Este material didático foi elaborado com fins educacionais. Solicitamos que qualquer erro encontrado ou dúvida com relação ao material ou seu uso seja enviado para a equipe de elaboração de conteúdo da Escola Superior de Redes, no e-mail [info@esr.rnp.br](mailto:info@esr.rnp.br). A Rede Nacional de Ensino e Pesquisa e os autores não assumem qualquer responsabilidade por eventuais danos ou perdas, a pessoas ou bens, originados do uso deste material.  
As marcas registradas mencionadas neste material pertencem aos respectivos titulares.

Distribuição  
**Escola Superior de Redes**

Rua Lauro Müller, 116 – sala 1103  
22290-906 Rio de Janeiro, RJ  
<http://esr.rnp.br>  
[info@esr.rnp.br](mailto:info@esr.rnp.br)

---

### Dados Internacionais de Catalogação na Publicação (CIP)

L799a      Lobo, Eduardo Augusto Oliveira  
             Administração de sistemas Linux: serviços para internet / Eduardo Augusto Oliveira  
             Lobo, Wagner Vieira Léo. – 2. ed. – Rio de Janeiro: RNP/ESR, 2013.  
             258 p. : il. ; 28 cm.

Bibliografia: p. 237-239.  
ISBN 978-85-63630-22-3

1. Linux (Sistema operacional de computador). 2. Serviços de diretório (tecnologia de redes de computador). I. Léo, Wagner Vieira. II. Título.

CDD 005.7

# Sumário

## **Escola Superior de Redes**

A metodologia da ESR   xiii

Sobre o curso   xiv

A quem se destina   xiv

Convenções utilizadas neste livro   xiv

Permissões de uso   xv

Sobre os autores   xvi

## **1. Kernel-based Virtual Machine – KVM**

Introdução à virtualização   1

Máquinas virtuais   2

    Máquinas virtuais de processo   2

    Monitores de máquinas virtuais   3

Virtualização total   3

Paravirtualização   4

    Vantagens e desvantagens da virtualização   5

Soluções de virtualização   5

O KVM: origem   6

    Arquitetura   6

    Gerenciamento de memória do Linux   7

    Exercício de nivelamento 1 – Gerenciamento de memória do Linux   7

    Gerenciamento de memória no KVM   7

    Exercício de fixação 1 – Pré-requisitos do hardware   9

    Exercício de fixação 2 – Instalação do KVM   9





|                                                           |           |
|-----------------------------------------------------------|-----------|
| Exercício de fixação 3 – Criando máquina virtual Linux    | 10        |
| Teclas de atalho                                          | 11        |
| Iniciando as máquinas virtuais                            | 12        |
| Interfaces de rede                                        | 13        |
| <b>Redes virtuais</b>                                     | <b>14</b> |
| Configurando uma rede privada entre máquinas virtuais     | 14        |
| Acessando a rede do host                                  | 15        |
| Automatizando a configuração de redes virtuais            | 15        |
| <b>Snapshots</b>                                          | <b>16</b> |
| Gerenciando snapshots                                     | 17        |
| <b>Roteiro de Atividades 1</b>                            | <b>19</b> |
| Atividade 1.1 – Gerenciando snapshots                     | 19        |
| Atividade 1.2 – Gerenciando a interface de rede           | 19        |
| Atividade 1.3 – Convertendo máquinas virtuais             | 20        |
| <b>2. Administração de máquinas virtuais com KVM</b>      |           |
| Virtualização no Linux                                    | 21        |
| Ferramentas para gerenciamento de máquinas virtuais       | 21        |
| QEMU-KVM                                                  | 22        |
| Eucalyptus                                                | 23        |
| OpenNebula                                                | 23        |
| Nimbus                                                    | 24        |
| oVirt                                                     | 24        |
| Plataforma em nuvem Xen                                   | 24        |
| OpenQRM                                                   | 24        |
| Virt-Manager                                              | 24        |
| Instalação do gerenciador gráfico Virt-Manager            | 24        |
| Exercício de fixação 1 – Instalando o pacote Virt-Manager | 25        |
| Criando uma máquina virtual                               | 25        |
| Acessando a máquina virtual                               | 26        |
| Configurando a interface de rede da máquina virtual       | 26        |
| Exercício de fixação 2 – Criando uma rede virtual         | 27        |
| Exercício de fixação 3 – Criando uma máquina virtual      | 27        |
| Exercício de fixação 4 – Importando uma máquina virtual   | 27        |
| Filtragem de pacotes                                      | 28        |

|                                                   |           |
|---------------------------------------------------|-----------|
| Hardware da máquina virtual                       | 28        |
| Placa de vídeo                                    | 29        |
| Detalhes da máquina virtual                       | 30        |
| Adicionando novo storage à máquina virtual        | 31        |
| Migração de máquinas virtuais                     | 32        |
| <br>                                              |           |
| <b>Roteiro de Atividades 2</b>                    | <b>35</b> |
| Atividade 2.1 – Live Migration                    | 35        |
| Atividade 2.2 – Criando clones                    | 42        |
| <br>                                              |           |
| <b>3. DNS e NFS</b>                               |           |
| Introdução                                        | 43        |
| Domain Name Service (DNS)                         | 44        |
| Por que utilizar DNS?                             | 44        |
| Definições                                        | 46        |
| Domínio                                           | 46        |
| Zonas de autoridade                               | 47        |
| Exercício de fixação 1 – DNS authoritative answer | 48        |
| Registro de recursos                              | 48        |
| Mapeamento direto e reverso                       | 51        |
| Cliente e servidor DNS                            | 52        |
| Servidor DNS                                      | 52        |
| Tipos de servidores DNS                           | 53        |
| Servidores raiz DNS                               | 55        |
| Estrutura do DNS no Linux                         | 55        |
| Arquivo de configuração ‘named.conf’              | 56        |
| Arquivos de mapas da rede                         | 58        |
| Exemplo de arquivo de mapa de rede                | 60        |
| Arquivo ‘resolv.conf’                             | 60        |
| Exercício de fixação 2 – Servidor de DNS Cache    | 61        |
| DNS com IPv6                                      | 61        |
| DNSSEC                                            | 63        |
| Implementação do DNSSEC                           | 66        |
| Exercício de fixação 3 – DNSSec                   | 67        |
| Domínios virtuais                                 | 67        |
| Testando o servidor DNS                           | 68        |



## **Network File System (NFS) 68**

Configurando um servidor NFS 69

Exercício de fixação 4 – Instale o servidor NFS 69

/etc/exports 70

/etc/hosts.allow e /etc/hosts.deny 71

## **Iniciando os serviços NFS 72**

Pré-requisitos 73

Iniciando o portmapper 73

Verificando a execução do NFS 73

Configuração de cliente NFS 74

Montagem de sistemas de arquivos NFS em tempo de boot 75

Otimizando o desempenho do NFS 76

## **Tamanho de pacote e drivers de rede 78**

## **Overflow de pacotes fragmentados 78**

NFS sobre TCP 79

## **NFS síncrono versus assíncrono 80**

## **Segurança e NFS 81**

NFS e segurança: portmapper 82

NFS para MS Windows 82

## **Roteiro de Atividades 3 85**

Atividade 3.1 – Servidor de DNS Primário 85

Atividade 3.2 – Servidor de DNS Secundário 85

Atividade 3.3 – Exportando diretório do servidor NFS 85

Atividade 3.4 – Configuração do Cliente NFS 85

Atividade 3.5 – Testando o funcionamento do serviço NFS 85

## **4. Servidor LDAP**

Introdução 87

LDAP 87

Sem serviço de diretório 88

Com serviço de diretório 88

Funcionamento do LDAP 89

Organização dos dados 91

Exercício de fixação 1 – Estrutura de diretório LDAP 91

## **Tipos de dados e unidade básica de informação 91**

Referência e autenticação entre vários servidores 93



|                                                                                           |            |
|-------------------------------------------------------------------------------------------|------------|
| Replicação                                                                                | 94         |
| Objetos e atributos LDAP                                                                  | 95         |
| Formato LDIF                                                                              | 95         |
| Classes LDAP                                                                              | 96         |
| Instalação do servidor LDAP                                                               | 96         |
| Download do pacote                                                                        | 97         |
| Desempacotando o software                                                                 | 97         |
| Configuração do software                                                                  | 98         |
| Construindo o servidor                                                                    | 98         |
| Configuração do servidor LDAP                                                             | 99         |
| Executando o servidor LDAP                                                                | 103        |
| Criação e manutenção de base de dados                                                     | 104        |
| Criando uma base de dados on-line                                                         | 104        |
| Criando base de dados off-line                                                            | 105        |
| Informações e características adicionais                                                  | 106        |
| <br>                                                                                      |            |
| <b>Roteiro de Atividades 4</b>                                                            | <b>107</b> |
| Atividade 4.1 – Instalação do servidor OpenLDAP                                           | 107        |
| Atividade 4.2 – Usando o Migration Tools                                                  | 108        |
| Atividade 4.3 – Configuração do cliente Linux para uso do LDAP                            | 109        |
| Atividade 4.4 – Criação e teste de base de dados para ser adicionada ao servidor OpenLDAP | 111        |
| Atividade 4.5 – Testando o funcionamento do LDAP                                          | 112        |
| <br>                                                                                      |            |
| <b>5. DHCP, Telnet, FTP e SSH</b>                                                         |            |
| Introdução                                                                                | 113        |
| DHCP                                                                                      | 114        |
| Formato de uma mensagem DHCP                                                              | 115        |
| Funcionamento do protocolo DHCP                                                           | 116        |
| Edição de arquivos de configuração                                                        | 118        |
| Base de dados                                                                             | 119        |
| Iniciando e parando o servidor                                                            | 121        |
| Exercício de fixação 1 – Instalação do servidor DHCP                                      | 122        |
| Telnet                                                                                    | 122        |
| Cliente Telnet                                                                            | 122        |
| Servidor Telnet                                                                           | 123        |

## **File Transfer Protocol (FTP) 125**

Funcionamento do FTP 126

Tipos de servidor FTP 126

## **Estrutura de diretórios típica 127**

Problemas com firewall 129

## **Secure Shell (SSH) 129**

OpenSSH 130

Configuração do servidor OpenSSH 130

Comando *ssh* 131

Estabelecendo conexão 132

SCP 132

SFTP 133

Geração de chaves 133

Agente SSH 134

## **Roteiro de Atividades 5 135**

Atividade 5.1 – Configure o servidor DHCP 135

Atividade 5.2 – Configuração de um endereço IP fixo 136

Atividade 5.3 – Utilização de redes compartilhadas 136

Atividade 5.4 – Serviço Telnet 137

Atividade 5.5 – Serviço FTP 137

Atividade 5.6 – Criação de uma conexão segura a um host remoto 137

Atividade 5.7 – Automatizar a criação de conexão segura com o servidor remoto por meio do compartilhamento da chave pública 138

Atividade 5.8 – Acesso de grupos a diretório compartilhado no FTP 138

Atividade 5.9 – Transferência de um arquivo de um host remoto para o host local e vice-versa 138

Atividade 5.10 – Criando uma conexão FTP segura com o comando SFTP 138

## **6. Servidor web**

Introdução 139

Conceitos fundamentais 139

Campos da URL 140

Esquema de funcionamento 142

Protocolo HTTP 142

Tipos de pedidos 143



|                                                                                    |            |
|------------------------------------------------------------------------------------|------------|
| Servidor web – proxy                                                               | 145        |
| Domínio Virtual                                                                    | 145        |
| Secure Socket Layer (SSL)                                                          | 146        |
| Servidores web                                                                     | 147        |
| Servidor web Apache                                                                | 147        |
| Exercício de fixação 1 – Correlacione as colunas                                   | 149        |
| <b>Roteiro de Atividades 6</b>                                                     | <b>151</b> |
| Atividade 6.1 – Instalação do servidor web Apache                                  | 151        |
| Atividade 6.2 – Configuração de servidores virtuais                                | 151        |
| Atividade 6.3 – Configuração de um servidor seguro                                 | 151        |
| Atividade 6.4 – Usando <i>.htaccess</i> para definir permissão de acesso a um site | 152        |
| Atividade 6.5 – Habilitando páginas pessoais de usuários                           | 152        |
| <b>7. Servidor de correio eletrônico (parte 1)</b>                                 |            |
| Introdução                                                                         | 153        |
| Correio eletrônico                                                                 | 153        |
| Protocolo SMTP                                                                     | 155        |
| Protocolo POP3                                                                     | 157        |
| Comandos POP3                                                                      | 157        |
| Protocolo IMAP                                                                     | 159        |
| Conexão e estados do IMAP                                                          | 160        |
| Configurações prévias do IMAP                                                      | 162        |
| DNS                                                                                | 163        |
| Postfix                                                                            | 165        |
| Instalação do servidor Postfix                                                     | 166        |
| Inicialização do servidor Postfix                                                  | 167        |
| Configuração do servidor Postfix                                                   | 168        |
| <b>Roteiro de Atividades 7</b>                                                     | <b>171</b> |
| Atividade 7.1 – Instalação e configuração do Postfix                               | 171        |
| Atividade 7.2 – Enviando e recebendo mensagens via Telnet                          | 174        |
| Atividade 7.3 – Enviando e recebendo mensagens                                     | 175        |

## **8. Servidor de correio eletrônico (parte 2)**

Introdução 177

Métodos de entrega 177

mbox 178

maildir 178

Servidor para múltiplos domínios 179

Domínios virtuais 179

Controle de conteúdo 181

Mail gateway 182

Autenticação SMTP 183

Servidor de e-mail corporativo 185

Cyrus SASL 186

Courier Maildrop 187

Courier Imap 187

Recomendações de tuning 188

Ajustes em consultas DNS 188

**Roteiro de Atividades 8 193**

Atividade 8.1 – Configuração da modalidade de entrega maildir 193

Atividade 8.2 – Configurando servidores POP e IMAP 193

Atividade 8.3 – Configurando a autenticação POP e IMAP no LDAP 194

Atividade 8.4 – Utilização de clientes POP e IMAP 195

## **9. Servidor proxy Squid**

Introdução 197

Proxy 198

Proxy Squid 199

Instalação do Squid 200

Configuração do Squid 200

Listas de controle de acesso 201

Configuração dos navegadores 203

Proxy transparente 204

Redirecionadores 205

## **Roteiro de Atividades 9 209**

Atividade 9.1 – Instalação e configuração do servidor proxy Squid 209

Atividade 9.2 – Configuração dos navegadores 212

Atividade 9.3 – Configuração de listas de controle de acesso 212

Atividade 9.4 – Configuração do SARG 212

Atividade 9.5 – Proxy transparente 212

## **10. Servidor Samba**

Introdução ao Samba 213

Samba 214

Samba e projetos relacionados 215

Instalando o Samba 216

Configurando o Samba 218

Seção [homes] 219

Compartilhamento de um disco Linux 220

Como acessar o Windows do Linux 221

Montagem de diretórios do Windows 221

Configuração do Samba com SWAT 222

Servidor primário de domínio 223

Configurando o PDC Samba 223

Criação das contas de relação de confiança 224

Criação manual de contas de relação de confiança 224

Criação dinâmica de contas de relação de confiança 225

Adição de clientes no domínio 225

## **Roteiro de Atividades 10 227**

Atividade 10.1 – Configuração do servidor Samba 227

Atividade 10.2 – Testando o funcionamento do Samba 234

Atividade 10.3 – Windows no domínio do Samba 234

Atividade 10.4 – Compartilhamentos 234

Atividade 10.5 – Excluindo usuários 235

Atividade 10.6 – Conhecendo o SWAT 235

## **Bibliografia 237**





# Escola Superior de Redes

A Escola Superior de Redes (ESR) é a unidade da Rede Nacional de Ensino e Pesquisa (RNP) responsável pela disseminação do conhecimento em Tecnologias da Informação e Comunicação (TIC). A ESR nasce com a proposta de ser a formadora e disseminadora de competências em TIC para o corpo técnico-administrativo das universidades federais, escolas técnicas e unidades federais de pesquisa. Sua missão fundamental é realizar a capacitação técnica do corpo funcional das organizações usuárias da RNP, para o exercício de competências aplicáveis ao uso eficaz e eficiente das TIC.

A ESR oferece dezenas de cursos distribuídos nas áreas temáticas: Administração e Projeto de Redes, Administração de Sistemas, Segurança, Mídias de Suporte à Colaboração Digital e Governança de TI.

A ESR também participa de diversos projetos de interesse público, como a elaboração e execução de planos de capacitação para formação de multiplicadores para projetos educacionais como: formação no uso da conferência web para a Universidade Aberta do Brasil (UAB), formação do suporte técnico de laboratórios do Proinfo e criação de um conjunto de cartilhas sobre redes sem fio para o programa Um Computador por Aluno (UCA).

## A metodologia da ESR

A filosofia pedagógica e a metodologia que orientam os cursos da ESR são baseadas na aprendizagem como construção do conhecimento por meio da resolução de problemas típicos da realidade do profissional em formação. Os resultados obtidos nos cursos de natureza teórico-prática são otimizados, pois o instrutor, auxiliado pelo material didático, atua não apenas como expositor de conceitos e informações, mas principalmente como orientador do aluno na execução de atividades contextualizadas nas situações do cotidiano profissional.

A aprendizagem é entendida como a resposta do aluno ao desafio de situações-problema semelhantes às encontradas na prática profissional, que são superadas por meio de análise, síntese, julgamento, pensamento crítico e construção de hipóteses para a resolução do problema, em abordagem orientada ao desenvolvimento de competências.

Dessa forma, o instrutor tem participação ativa e dialógica como orientador do aluno para as atividades em laboratório. Até mesmo a apresentação da teoria no início da sessão de aprendizagem não é considerada uma simples exposição de conceitos e informações. O instrutor busca incentivar a participação dos alunos continuamente.

As sessões de aprendizagem onde se dão a apresentação dos conteúdos e a realização das atividades práticas têm formato presencial e essencialmente prático, utilizando técnicas de estudo dirigido individual, trabalho em equipe e práticas orientadas para o contexto de atuação do futuro especialista que se pretende formar.

As sessões de aprendizagem desenvolvem-se em três etapas, com predominância de tempo para as atividades práticas, conforme descrição a seguir:

**Primeira etapa: apresentação da teoria e esclarecimento de dúvidas (de 60 a 90 minutos).**

O instrutor apresenta, de maneira sintética, os conceitos teóricos correspondentes ao tema da sessão de aprendizagem, com auxílio de slides em formato PowerPoint. O instrutor levanta questões sobre o conteúdo dos slides em vez de apenas apresentá-los, convidando a turma à reflexão e participação. Isso evita que as apresentações sejam monótonas e que o aluno se coloque em posição de passividade, o que reduziria a aprendizagem.

**Segunda etapa: atividades práticas de aprendizagem (de 120 a 150 minutos).**

Esta etapa é a essência dos cursos da ESR. A maioria das atividades dos cursos é assíncrona e realizada em duplas de alunos, que acompanham o ritmo do roteiro de atividades proposto no livro de apoio. Instrutor e monitor circulam entre as duplas para solucionar dúvidas e oferecer explicações complementares.

**Terceira etapa: discussão das atividades realizadas (30 minutos).**

O instrutor comenta cada atividade, apresentando uma das soluções possíveis para resolvê-la, devendo ater-se àquelas que geram maior dificuldade e polêmica. Os alunos são convidados a comentar as soluções encontradas e o instrutor retoma tópicos que tenham gerado dúvidas, estimulando a participação dos alunos. O instrutor sempre estimula os alunos a encontrarem soluções alternativas às sugeridas por ele e pelos colegas e, caso existam, a comentá-las.

## Sobre o curso

O curso tem como objetivo ensinar a projetar, instalar, configurar e disponibilizar os principais serviços para internet em uma rede TCP/IP. Apresenta os conceitos associados a cada um dos serviços, além da instalação e configuração do KVM como base para o ambiente de virtualização, e autenticação nos serviços com LDAP, com apoio intensivo de atividades práticas.

## A quem se destina

Esse curso é destinado à formação de profissionais responsáveis pela instalação, operação e manutenção de plataforma computacional para conexão com a internet, gerentes de TI e programadores que tenham os pré-requisitos necessários.

## Convenções utilizadas neste livro

As seguintes convenções tipográficas são usadas neste livro:

*Itálico*

Indica nomes de arquivos e referências bibliográficas relacionadas ao longo do texto.

Largura constante

Indica comandos e suas opções, variáveis e atributos, conteúdo de arquivos e resultado da saída de comandos. Comandos que serão digitados pelo usuário são grifados em negrito e possuem o prefixo do ambiente em uso (no Linux é normalmente # ou \$, enquanto no Windows é C:\).

**Conteúdo de slide** 

Indica o conteúdo dos slides referentes ao curso apresentados em sala de aula.

**Símbolo** 

Indica referência complementar disponível em site ou página na internet.

**Símbolo** 

Indica um documento como referência complementar.

**Símbolo** 

Indica um vídeo como referência complementar.

**Símbolo** 

Indica um arquivo de áudio como referência complementar.

**Símbolo** 

Indica um aviso ou precaução a ser considerada.

**Símbolo** 

Indica questionamentos que estimulam a reflexão ou apresenta conteúdo de apoio ao entendimento do tema em questão.

**Símbolo** 

Indica notas e informações complementares como dicas, sugestões de leitura adicional ou mesmo uma observação.

## Permissões de uso

Todos os direitos reservados à RNP.

Agradecemos sempre citar esta fonte quando incluir parte deste livro em outra obra.

Exemplo de citação: LOBO, Eduardo Augusto Oliveira; LÉO, Wagner Vieira. *Administração de Sistemas Linux: Serviços para Internet*. Rio de Janeiro: Escola Superior de Redes, RNP, 2013.

## Comentários e perguntas

Para enviar comentários e perguntas sobre esta publicação:

Escola Superior de Redes RNP

Endereço: Av. Lauro Müller 116 sala 1103 – Botafogo

Rio de Janeiro – RJ – 22290-906

E-mail: [info@esr.rnp.br](mailto:info@esr.rnp.br)

## Sobre os autores

**Eduardo Lobo** Analista de Informática no Ministério Público do Trabalho (MPT) e Professor dos Cursos de TI da Universidade Católica de Brasília. Engenheiro Mecânico graduado pela Universidade Federal de Uberlândia e Mestre em Ciência da Computação pela Universidade Federal de Santa Catarina, Florianópolis, 2001. Atuou em grandes empresas no período de 1997-2000, como Analista de TI para o governo brasileiro em 2001, e como Coordenador dos Cursos de Pós-Graduação Lato Sensu em Redes de Computadores da Universidade Católica de 2003 a 2005, e de Tecnologia em Segurança da Informação (TSI) de 2007 a 2009. Participou do Projeto de Pesquisa CESMIC/UCB de 2003 a 2008. Atualmente administra o datacenter e a rede de dados/voz do MPT. Áreas de interesse incluem: redes de computadores, storage e sistemas baseados em software livre.

**Wagner Vieira Léo** tem 25 anos de experiência na área de TI, atuando como Analista de Suporte e em Computação de Alto Desempenho, com foco em sistemas operacionais Unix/Linux. Possui graduação em Matemática pela Faculdade de Humanidades Pedro II, com Pós-Graduação em Gestão da Inovação pelo LNCC/UCP. Atualmente ocupa o cargo de Coordenador de Tecnologia da Informação do Laboratório Nacional de Computação Científica e de Coordenador Administrativo do Ponto de Presença da RNP no Rio de Janeiro. Professor do Instituto Superior de Tecnologia da Informação de Petrópolis, IST e da Escola Superior de Redes da RNP, nas áreas de Segurança da Informação e Sistemas Operacionais.

**Bruno Alves Fagundes** é tecnólogo em Tecnologia da Informação e da Computação pelo Instituto Superior de Tecnologia (2007) e especialista em Segurança de Redes pela Universidade Estácio de Sá (2011). Tem experiência em administração de servidores Linux, gerenciamento de usuário, implementação e gerência de serviços de rede, clusters, segurança de redes, shell script, PHP, Perl. Atualmente é colaborador do LNCC, onde trabalha na administração dos servidores e clusters do Laboratório de Bioinformática (LABINFO), provendo infraestrutura para a realização de pesquisas e processamento massivo de dados.

**Francisco Marcelo M. Lima** é certificado Project Management Professional (PMP) e Modulo Certified Security Officer (MCSO), Mestre em Engenharia Elétrica pela Universidade de Brasília (2009), Mestre em Liderança pela Universidade de Santo Amaro (2007) e pós-graduado em Segurança de Redes de Computadores pela Universidade Católica de Brasília (2003). Atualmente exerce as funções de Coordenador dos Cursos de Redes de Computadores e Segurança da Informação do IESB, e Analista em TI do MPOG cedido para a Controladoria-Geral da União/PR. Atua também como instrutor/revisor dos cursos de segurança e redes na ESR e instrutor/revisor dos cursos de planejamento estratégico (PDTI) e gestão de contratos de TI (GCTI) na ENAP. Possui mais de 15 anos de experiência na área de Ciência da Computação, com ênfase em Segurança da Informação, Redes e Construção de Software, tendo exercido funções como: Coordenador Geral de TI do INCRA (DAS 4); Coordenador do Curso de Segurança da Informação da Faculdade Rogacionista; Coordenador do Curso de Processamento de Dados e Segurança da Informação da Faculdade AD1, Analista em Segurança da empresa Módulo Security Solutions.

**Sergio Ricardo Alves de Souza** possui mais de 35 anos de experiência na área de Administração e Suporte de Sistemas. Trabalhou em empresas como: Burroughs (UNISYS), ARSA (Infraero), Cobra Computadores, LNCC e outras. Consultoria e treinamento em empresas como: Fiocruz, Jardim Botânico, Museu Goeldi, Cefet-MG, IBM Brasil. Participou do desenvolvimento e implantação de cursos profissionalizantes em Informática em Petrópolis - FAETC. Participou do projeto de criação do Instituto Superior de Tecnologia em Ciência da Computação de Petrópolis. Possui "Notório Saber em Informática" pelo LNCC.

# 1

## Kernel-based Virtual Machine – KVM

### objetivos

Entender o que é virtualização e o funcionamento de máquinas virtuais, os principais tipos de hipervisores e uma alternativa open source de virtualização, o KVM.

Arquitetura e funcionamento do KVM, métodos operacionais como criação de discos, snapshots, conversão do tipo de imagem e criação de máquinas virtuais.

### conceitos

### Introdução à virtualização

- Surgiu nos mainframes da IBM e ficou um pouco esquecida com o crescimento dos computadores de menor porte.
- Arquitetura x86 apresentava dificuldades na implementação da virtualização.
- VMware surge com um conceito eficiente de virtualização para arquitetura x86.
- AMD e Intel melhoram o suporte à virtualização de seus processadores, criando as tecnologias AMD-V e Intel VT.



Apesar de muito badalada, a virtualização não é uma novidade no mundo da Tecnologia da Informação. A ideia surgiu com a publicação de um artigo em 1959, escrito por Christopher Strachey, que tratava do uso de multiprogramação em tempo compartilhado, que fazia com que as máquinas utilizassem melhor os seus recursos. Esse artigo deu origem ao padrão Compatible Time Sharing System (CTSS), desenvolvido pelo MIT, e era usado como referência na fabricação de máquinas de grande porte.

A IBM surgiu com as primeiras formas de virtualização quando introduziu em seus mainframes o conceito de multiprocessamento e memória virtual, o que permitia que várias CPUs trabalhassem como uma só e ainda possibilitava uma abstração da memória real para a memória virtual. Em sua linha dos mainframes 370 e sucessores, a IBM oferecia uma máquina virtual portada para várias de suas plataformas.

Devido ao elevado preço dos mainframes e a popularização dos computadores de arquitetura x86 (décadas de 80 e 90), muitas empresas passaram a comprar vários computadores de menor porte para fazer o trabalho dos mainframes.



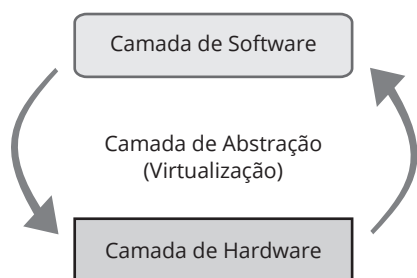
Como as máquinas de arquitetura x86 não eram projetadas para oferecer serviço de virtualização eficiente e sempre que se propunham a isso, acabavam superdimensionando os servidores, o que remetia à época dos mainframes. Foi então que a VMware, em 1999, surgiu com o conceito de virtualização mais eficiente para a plataforma x86 e começou a mudar o mercado. Posteriormente, em 2005, a AMD e a Intel melhoraram o suporte à virtualização em seus processadores, criando as tecnologias AMD-V e Intel VT.

## Máquinas virtuais

Uma máquina virtual pode ser definida como uma camada de abstração entre a camada de hardware e a camada de software. Essa camada de abstração permite o isolamento entre as outras camadas.

- Máquinas Virtuais de Processo.
- Monitores de Máquinas Virtuais (VMM) ou hipervisores.
  - ▣ Paravirtualização.
  - ▣ Virtualização total.

A máquina virtual pode ser definida como uma camada de abstração entre a camada de hardware e a camada de software (Figura 1.1). Essa camada de abstração permite o isolamento entre as outras camadas.



**Figura 1.1**  
Camada de abstração.

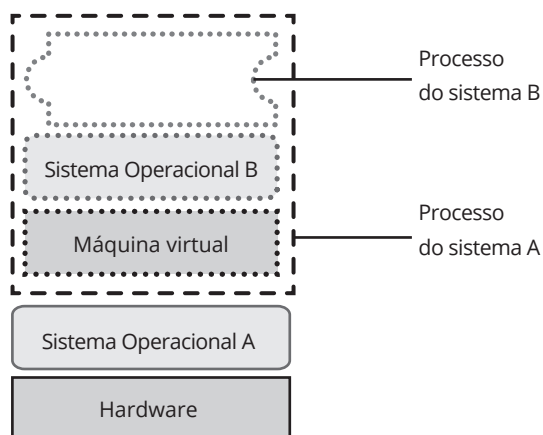
## Máquinas virtuais de processo

Uma máquina virtual de processo nada mais é do que uma aplicação que executa sobre um Sistema Operacional A e emula o comportamento de um Sistema Operacional B. Assim, aplicações desenvolvidas para um Sistema Operacional B podem executar sobre um sistema A (Figura 1.2).

Nesse modelo, o desempenho das máquinas virtuais é sacrificado, já que existe uma tradução de um sistema para outro, uma vez que as máquinas executam em modo de usuário e não têm acesso direto a alguns recursos do sistema.



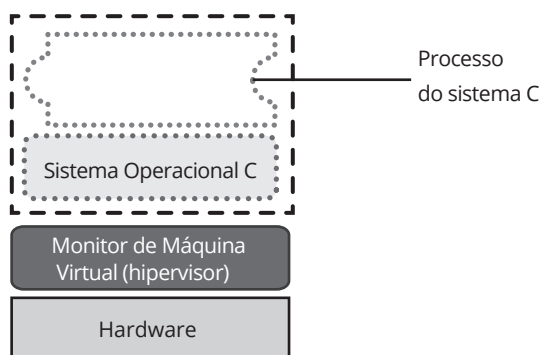
**Figura 1.2**  
Máquina virtual  
de processo.



## Monitores de máquinas virtuais

Implementados como uma camada de software entre o hardware e o Sistema Operacional, os monitores de máquinas virtuais, também chamados de hipervisores (hypervisors), surgem para resolver as desvantagens das máquinas virtuais de processo. Como possuem acesso direto ao hardware, conseguem explorar os recursos físicos de E/S de maneira mais eficiente. O desempenho tende a ser melhor, pois não executam em modo usuário (Figura 1.3).

**Figura 1.3**  
Monitores de  
Máquinas Virtuais.



Na arquitetura x86, os privilégios de acesso ao hardware e às operações de sistema são divididos em quatro níveis (nível 0, 1, 2 e 3), com o objetivo de oferecer mais segurança para o Sistema Operacional. Esses níveis são chamados de anéis ou Rings e estão organizados de forma hierárquica: o nível 0 é o de maior privilégio e o 3, o nível com mais restrições. As aplicações dos usuários rodam no nível 3, enquanto o Sistema Operacional roda no nível 0 com o seu kernel controlando o acesso ao hardware.

Apesar da segurança oferecida pela arquitetura x86, essa implementação apresenta desafios para os sistemas virtualizados, já que a camada de virtualização precisa ter acesso aos recursos do hardware no nível 0. Existem duas técnicas utilizadas nos hipervisores para permitir esse acesso: virtualização total e paravirtualização.

## Virtualização total

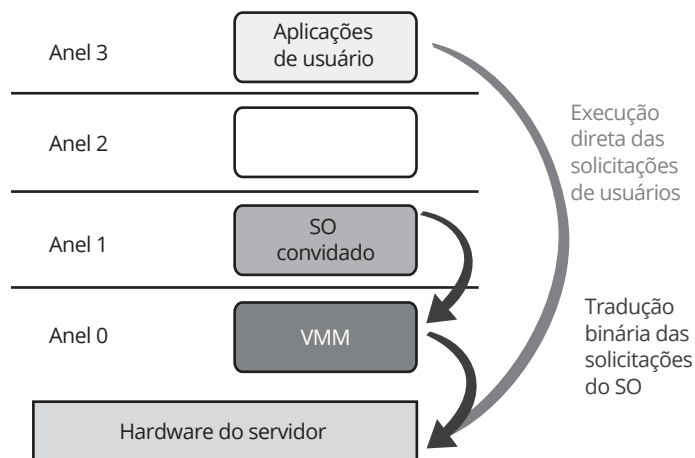
- É a técnica que permite executar qualquer sistema sem a necessidade de alteração.
- Nessa técnica, todo o hardware da máquina é simulado, assim como o conjunto de instruções do processador.



- Geralmente, os dispositivos disponibilizados para as máquinas virtuais são padrão de mercado, o que torna muito mais simples a instalação e configuração dos sistemas guest.



Como podemos observar na Figura 1.4, o guest está rodando em um nível superior ao nível esperado (Ring0). Isso porque o hipervisor fornece todo o ambiente para a máquina virtual “acreditar” que está rodando diretamente no nível mais privilegiado.



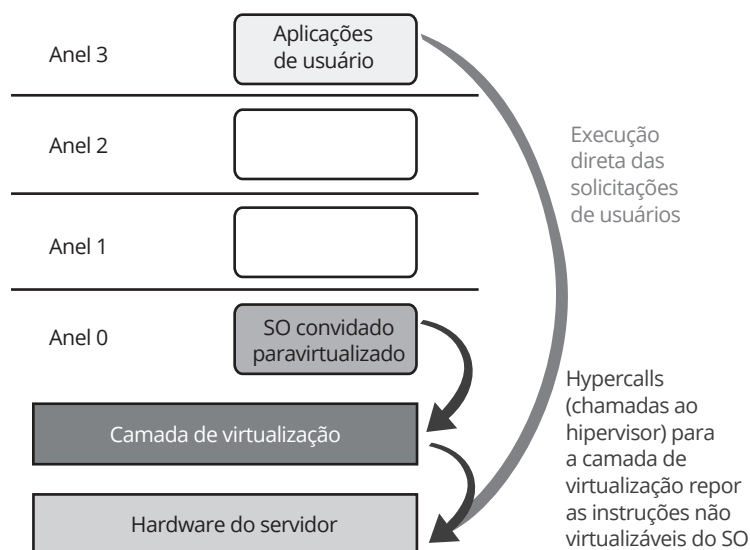
**Figura 1.4**  
Virtualização total.

## Paravirtualização

- A paravirtualização apresenta uma Application Programming Interface (API) muito semelhante ao hardware real.
- Para as máquinas virtuais, ela é responsável pela comunicação entre o sistema guest e o hipervisor.
- Para isso foram criadas as hypercalls, que são system calls otimizadas para o hipervisor.



Dessa forma, é fornecida uma interface de comunicação para o gerenciamento de memória, controle de interrupções e operações críticas do kernel para as máquinas virtuais. São necessárias mudanças no Sistema Operacional do host para usar utilizar as hypercalls.



**Figura 1.5**  
Paravirtualização.

## Vantagens e desvantagens da virtualização

Vantagens:

- Gerenciamento Centralizado.
- Facilidade de manutenção (clonagem e snapshot).
- Melhor aproveitamento do espaço físico.
- Melhor aproveitamento do hardware.
- Reduz os índices de emissão de CO2.
- Aumento da disponibilidade (live migration).
- Paradigma de um servidor por serviço.

O principal apelo para aderir à virtualização é a economia oferecida pela solução: estima-se que a utilização dos servidores fique entre 2% e 15% do seu potencial máximo, mas existem outros aspectos interessantes que podem pesar na hora de optar pela infraestrutura de virtualização.

Existem alguns cuidados que devem ser tomados para evitar que a solução de virtualização se torne um problema. Serão necessárias medidas que vão desde a redundância dos equipamentos até a escolha do software ideal para gerenciar as máquinas virtuais.

Desvantagens:

- Custos iniciais elevados de implementação.
- Necessidade de mão de obra qualificada.
- Muitos serviços em um mesmo hardware.
- Número desnecessário de máquinas virtuais.
- Desempenho comprometido.

Não é qualquer aplicação que deve ser virtualizada. Aplicações com muito I/O ou que consomem muita largura banda não devem ser virtualizadas, pois podem acabar comprometendo o desempenho das outras máquinas virtuais. Esses problemas podem ser sanados com a aquisição de mais hardware, CPU e placa de rede, mantendo o resto da estrutura.

## Soluções de virtualização

Principais soluções de virtualização no mercado:

- VMware ESX.
- Hyper-V.
- Xen.
- KVM.

### VMware ESX

Possui poderosa infraestrutura e oferece muitos recursos avançados para gerenciamento e administração de um ambiente virtual, como VMotion (migração de máquinas virtuais), recursos de alta disponibilidade, tolerância a falhas, Storage Motion, entre outros. Mesmo sendo considerada uma solução de custo elevado, é o principal software usado por grandes empresas.

### Hyper-V

É a solução oferecida pela Microsoft para sistemas de virtualização. É utilizado principalmente em empresas de pequeno e médio porte e possui recursos como Live Migration,



Quick Migration e Dinamic Memory. É oferecido de forma gratuita, entretanto é necessário ter a licença do Windows Server para usá-lo.

## Xen

Baseado em Linux, é menos popular que o Hyper-V e o VMware ESX, mas é uma solução muito estável. Fornece interface de administração por linha de comando (CLI), além de uma alternativa gráfica (GUI): XenClient – essa é a melhor indicação para quem já utiliza produtos da Citrix (empresa que atua na área de virtualização, colaboração e **cloud**).

## KVM

O KVM aparece como alternativa completa para sistemas de virtualização baseados em Linux. Assim como o Xen, fornece administração por linha de comando e por interface gráfica, dispõe de recursos de migração online e offline, possui o sistema de otimização do gerenciamento de memória Kernel Samepage Merging (KSM), utiliza o protocolo QEMU Monitor Protocol (QMP); que é um protocolo Java Script Object Notation (JSON-based), é open source e permite adicionar CPUs e dispositivos PCI em tempo real.

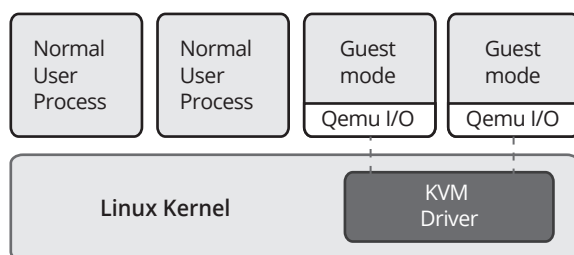
## O KVM: origem

O Kernel-based Virtual Machine (KVM) foi inicialmente desenvolvido pela empresa Qumranet entre 2006 e 2007 e teve seu crescimento mais acentuado, em 2008, quando a empresa foi adquirida pela RedHat, que a partir de então incorporou o KVM às suas distribuições.

O KVM é uma solução completa de virtualização para Linux com arquitetura x86 de 64 bits, e explora as tecnologias de virtualização Intel VT e AMD-V dos processadores mais recentes. É distribuído como um componente integrado do kernel a partir da versão 2.6.20 e transforma o Linux em um hipervisor estável e de alto desempenho.

## Arquitetura

Ao carregar o módulo do KVM no Linux, o kernel exporta um dispositivo chamado `/dev/kvm`, que fornece ao sistema um novo modo de execução, além dos modos já existentes kernel e user, chamado de guest mode (Figura 9.6). Todos os processos no espaço de usuário acessam de forma compartilhada os dispositivos do `/dev`, entretanto, o dispositivo `/dev/kvm` permite que uma máquina virtual tenha seu espaço de endereço independente do kernel ou de qualquer outra máquina virtual, permitindo o seu isolamento, ou seja, cada processo que acessa o dispositivo `/dev/kvm` recebe um mapa de endereços diferentes.



**Figura 1.6**  
Arquitetura KVM.

A estrutura do KVM é dividida em dois componentes básicos:

- **KVM driver:** é um conjunto de módulos do kernel que fornecem os drivers específicos para o tipo de processador e a infraestrutura necessária para a virtualização, atuando como interface para o user space;

- **User space:** roda uma versão modificada do qemu, que gerencia E/S e funciona como um espaço virtual para os sistemas hóspedes, além de fornecer mecanismos de controle e gerenciamento das máquinas virtuais;

As máquinas virtuais são tratadas como processos do Sistema Operacional, e por isso é possível usar os comandos do sistema para manipulá-las. Junto com cada máquina, é instanciado um processo do qemu, levemente modificado, que intercepta qualquer solicitação de I/O feita pela máquina e a emula. Além das solicitações de I/O, o qemu é responsável por emular os dispositivos, como adaptadores gráficos, dispositivos de rede, discos, entre outros.

- i440FX PCI bridge e PIIX3 ISA bridge;
- Cirrus CLGD 5446 PCI VGA;
- Mouse e teclado PS/2;
- Drive de disquete;
- Adaptadores de rede PCI e ISA;
- Portas seriais;
- Placa de som Creative Sound Blaster 16;
- PCI UHCI USB controller e Hub USB virtual.

A utilização do KVM, junto com o qemu, permite que as máquinas virtuais rodem com baixo overhead, tornando-o uma alternativa competitiva às opções do mercado.

## Gerenciamento de memória do Linux

O Linux é baseado no conceito de Memória Virtual e utiliza técnicas de paginação de memória, ou seja, quando um processo está rodando na máquina, não há alocação de toda a memória de uma única vez. Essa alocação é feita de acordo com a necessidade, tornando a utilização da memória mais eficiente.

Quando a memória física é totalmente utilizada ou um processo fica um determinado tempo sem ser utilizado, o Linux move a memória de um processo para o swap. Como o swap é feito em disco, torna-se significativamente mais lento que a RAM.

### Exercício de nivelamento 1

#### Gerenciamento de memória do Linux

Relacione as colunas:

- |                    |                                                                                         |
|--------------------|-----------------------------------------------------------------------------------------|
| 1. Memória física  | <input type="checkbox"/> Pode alocar grupos de páginas contíguas.                       |
| 2. Memória virtual | <input type="checkbox"/> Responsável pelo uso de endereçamento de cada processo.        |
|                    | <input type="checkbox"/> Gerencia a transferência de páginas entre o disco e a memória. |
|                    | <input type="checkbox"/> Responsável por alocar e desalocar páginas de memória.         |

## Gerenciamento de memória no KVM

O KVM usa todos os benefícios do gerenciamento de memória do Linux. Como as máquinas virtuais são processos para o host, o KVM pode alocar mais memória para os guests do que realmente existe no host. Não existe um bloco de memória física dedicado, os guests podem ser manipulados com comandos do Sistema Operacional. Utilizando o swap, é possível diminuir a quantidade de memória real necessária para os guests.

## Swapping

O host escolhe algumas páginas de memória do guest e envia para o disco. Se o guest necessitar daquela informação, o host trás de volta as páginas para a memória principal. É o método mais confiável do ponto de vista do host, entretanto, para o guest pode não ser a melhor escolha, porque o host não tem controle sobre os processos que estão em execução dentro do guest.

## Balão de memória

Mais eficiente do que o swapping do ponto de vista do guest, o balão de memória funciona de forma cooperativa: o guest se comunica com o host e informa quais páginas de memória podem ser liberadas. Pelo fato de a escolha ser feita pelo guest, não possui tanta confiabilidade.

É possível obter informações sobre esse mecanismo através do console do qemu. Exemplo:

```
(qemu) info balloon  
balloon: actual 1024
```

## Kernel Same-page Merging (KSM)

Realiza uma unificação das páginas de memória que são iguais. Dessa forma, as páginas duplicadas são removidas e seu espaço, liberado. Esse sistema oferece mais vantagem quando existirem muitas máquinas virtuais homogêneas.

Para melhorar o processo de gerenciamento de memória, foi desenvolvido um daemon chamado ksmtd, que gerencia dinamicamente os ajustes de memória do KSM, tomando como base a quantidade de memória livre disponível.

## Live migration

Consistem em mover um guest de um host para outro. Dessa forma, libera a memória que estava sendo utilizada.

O KVM usa todas as técnicas para conseguir os melhores resultados no gerenciamento de memória. Preferencialmente são utilizados os métodos de balão de memória e KSM, em seguida o live migration e, em último caso, o swap.

## Sistemas convidados suportados

O KVM está em constante aperfeiçoamento e desenvolvimento. A lista dos Sistemas Operacionais convidados suportados está disponível na web.

## Pré-requisitos do sistema

Conforme mencionado anteriormente, para que o KVM funcione, é necessário que você esteja utilizando um processador com suporte à tecnologia de virtualização AMD-V ou Intel VT, versão do kernel igual ou superior a 2.6.20 e arquitetura de 64 bits.



Quando o total de memória usada pelos guests é maior do que a memória física disponível, temos um overcommit. Existem alguns mecanismos específicos para tratar o overcommit em sistemas de virtualização.



Acesse a lista atualizada em <http://linux-kvm.org>

Para verificar se o seu sistema tem suporte ao KVM, realize os seguintes passos:

1. Verificar compatibilidade do processador:

```
# egrep --color '(vmx|svm)' /proc/cpuinfo

flags              : fpuvme de psetscmsrpaemce cx8
apicsepmttrpgemcacmovpat pse36 clflushdtsacpimmxfxsrsse sse2
sshttmpbesyscallnxdtsclmconstant_tscarch_perfmonebsbtsrep_
goodxtopologynonstop_tscaperfmpfnipclmulqddtes64 monitor
ds_cplvmx est tm2 ssse3 cx16 xtprpdc sse4_1 sse4_2 x2apic
popcntaesxsavexlahf_lm ida arattpr_shadowvnmiflexpriorityeptvpid
```

2. Versão do Kernel e arquitetura:

```
# uname -a

Linux debian 2.6.32-5-amd64 #1 SMP Sun May 6 04:00:17 UTC 2012
x86_64 GNU/Linux
```

## Exercício de fixação 1

### Pré-requisitos do hardware

Verifique se o Sistema Operacional tem suporte para as instruções de virtualização dos processadores AMD e Intel.

Verifique a versão do kernel e a arquitetura.

### Instalação

■ Debian/Ubuntu:

```
# apt-get install kvm qemu-kvm qemu-utils
```

■ CentOS/Red Hat:

```
# yum install kvm kmod-kvm qemu
```

■ SUSE/OpenSuse:

```
# yast2 -i kvm qemu
```

## Exercício de fixação 2

### Instalação do KVM

Instale o KVM através do repositório do Debian.

### Criando imagens

Conforme já mencionamos, o KVM utiliza o qemu para realizar as tarefas de emulação da máquina virtual e criação das imagens dos discos. O comando utilizado para criar imagens de discos é o *qemu-img*, que está disponível no pacote *qemu-kvm* (para Debian).

Um recurso muito interessante do *qemu-img* é que, ao criarmos um arquivo de 10Gb, o comando cria um arquivo vazio, ou seja, quando executamos o comando para criar um arquivo de 10Gb, na verdade é criado um arquivo de aproximadamente 140Kb, que vai aumentando de tamanho conforme vamos gravando as informações dentro dele. Isso se torna um recurso muito interessante quando não dispomos de muito espaço em disco no momento em que estamos criando as máquinas. É possível criar um disco que, no futuro, ocupará espaço maior do que o espaço atualmente disponível.

Também é possível criar um arquivo de disco com o comando *dd*. No exemplo a seguir criaremos um arquivo com 10Gb.

```
# dd if=/dev/zero of=disk.img bs=10G count=5
```

Quando utilizamos o comando *dd* para criar os discos, eles já possuem o tamanho total, enquanto que o arquivo criado com o comando *qemu-img* vai sendo ocupado conforme vamos utilizando o espaço disponível.

Formatos de imagens suportados:

- **raw**: é o formato padrão, facilmente exportado para outros emuladores.
- **cloop**: Linux Compressed Loop images. Útil apenas para imagens de CD-ROM compactadas.
- **cow**: copy-on-write. Existe apenas por razões históricas, não está disponível para Windows.
- **qcow**: antigo formato do qemu, atualmente substituído pelo formato qcow2.
- **qcow2**: formato atual do qemu, inclui suporte para múltiplos snapshots, encriptação AES e compressão utilizando a biblioteca zlib.
- **vmdk**: formato de imagem compatível com VMware 3 & 4, ou 6.
- **vdi**: formato de imagem compatível com VirtualBox 1.1.

Criando imagem no formato qcow2 com o nome de *winxp.img*, com tamanho máximo de 5Gb:

```
# qemu-img create -f qcow2 winxp.img 5G
```

Criando imagem compatível com o VirtualBox, com o tamanho máximo de 10Gb:

```
# qemu-img create -f vdi win7.vdi 10G
```

O comando *qemu-img* permite realizar a conversão de arquivos de imagem em outros formatos. É possível converter uma imagem no formato do VMware para o formato do KVM.

```
# qemu-imgconvert -f vmdkdebian.vmdk -O qcow2 debian.img
```

Mais opções de conversão podem ser encontradas acessando a *manpage* do comando:

```
# manqemu-img
```

## Exercício de fixação 3

### Criando máquina virtual Linux

1. No diretório */kvm/maquinas\_virtuais*, crie uma imagem de disco no formato qcow2 com o nome *linux.img* e tamanho máximo de 20Gb.
2. Inicie a máquina virtual informando o arquivo ISO de instalação do Linux, disponível para o curso.
3. Inicie a instalação do Sistema Operacional na máquina virtual.

## Utilização

É possível criar e configurar máquinas virtuais somente pela linha de comando. No KVM é possível modificar as configurações das máquinas apenas trocando alguns atributos. O KVM nos permite determinar a quantidade de memória, modelo das CPUs, drives de disquete, discos rígidos, unidade de CD-ROM, USB, unidades de disco SD, entre outros.

Os parâmetros básicos para utilizar o KVM são:

- **-m megas**: determina a quantidade de memória que será utilizada pela máquina virtual. Por padrão, o valor é expresso em Megabytes, mas é possível utilizar o sufixo G para informar a quantidade em Gigabytes. Se esse parâmetro for omitido, será reservado 128 MB para a máquina virtual;
- **-hd[a,b,c,d] arquivo**: especifica qual será o arquivo ou dispositivo que será carregado como unidade de disco da máquina virtual. É possível informar até quatro unidades de disco.
- **-cdrom arquivo**: especifica qual será o arquivo ou dispositivo que será montado como unidade de CD-ROM na máquina virtual. Não é permitido utilizar -hdc e -cdrom ao mesmo tempo.
- **-drive option[,options]**: define um novo drive, similar às opções -hd[a,b,c,d] e -cdrom. Principais opções do comando:
  - **file=filename**: define o arquivo de imagem que será utilizado.
  - **if=interface**: especifica o tipo de interface do disco: ide, scsi, sd, floppy, virtio etc.
  - **bus=bus,int=int**: especifica onde o dispositivo está conectado no host.
  - **snapshot=on|off**: ativa ou desativa o snapshot para a imagem.
- **-boot [order=drives][,onde=drives][,menu=on|off]**: informa a ordem do boot. A ordem padrão para sistemas de arquitetura x86 é: a, b (disquetes 1 e 2), c (primeira unidade de disco), d (primeiro CD-ROM) e n (boot pela interface de rede).
- **-g WxH[xDEPTH]**: modifica a resolução padrão que a máquina virtual iniciará. O padrão é 800x600x15.
- **-usb**: ativa o suporte para dispositivos USB.
- **-devicedriver[,option[,...]]**: adiciona um dispositivo na máquina virtual. As opções variam de acordo com o drive informado. Para listar os tipos de devices drives suportados pelo KVM, execute o comando `# kvm -device ?`

A relação completa das opções do comando está no manual do KVM (man kvm).

## Teclas de atalho

- **[Ctrl] + [Alt]**: libera o cursor do mouse e o teclado.
- **[Ctrl] + [Alt] + [u]**: retorna para a resolução padrão.
- **[Ctrl] + [Alt] + [f]**: modo fullscreen.
- **[Ctrl] + [Alt] + [1]**: mostra a saída de vídeo da máquina virtual.
- **[Ctrl] + [Alt] + [2]**: muda para o console de comando do KVM, o monitor.
- **[Ctrl] + [Alt] + [3]**: mostra a saída da porta serial.





## Iniciando as máquinas virtuais

Após a criação da imagem do disco, devemos iniciar nossa máquina virtual. Para isso, devemos utilizar o comando *KVM*:

```
# kvm -m 256 -hda winxp.img
```

O parâmetro *-m* informa a quantidade de memória que será alocada para a máquina virtual (em MB), no nosso exemplo, alocamos 256MB. Como nossa imagem está vazia, devemos instalar um Sistema Operacional. Para isso, podemos informar que a máquina acessará o drive de CD-ROM da máquina hospedeira. O parâmetro *-cdrom* informa para a máquina virtual qual será o dispositivo ou imagem que será usado como CD-ROM:

```
# kvm -cdrom /dev/cdrom -m 256 -hda winxp.img
# kvm -cdrom /home/aluno/windowsXP.iso -m 256 -hda winxp.img
```

Iniciando a máquina virtual com boot pelo CD-ROM:

```
# kvm -cdrom /dev/cdrom -m 256 -hda winxp.img -boot d
```

Para iniciar a máquina virtual com um dispositivo USB, primeiro devemos identificá-lo na máquina hospedeira. O comando *lsusb* informa o endereço físico do dispositivo.

```
# lsusb
[...]
Bus 002 Device 006: ID 0930:6544 Toshiba Corp. Kingston
DataTraveler2.0 ...
[...]
```

Agora usamos o parâmetro *-usb* para informar onde está o dispositivo.

```
# kvm -cdrom /dev/cdrom -m 256 -hda winxp.img -usb -device usb-
host,hostbus=2,hostaddr=6
```

Caso apareçam mensagens de erro de permissão no terminal quando a máquina virtual for iniciada, será necessário dar permissão para o usuário que está executando a máquina escrever no dispositivo USB. No exemplo a seguir optamos por dar permissão para outros usuários escreverem no dispositivo:

```
# chmod o+w /dev/bus/usb/002/006
```

Como todos os parâmetros para iniciar a máquina virtual devem ser passados pela linha de comando, uma boa prática é criar um arquivo Shell com esses parâmetros.

Vamos criar um script para iniciar uma máquina Windows, com 512Mb de memória RAM, dois discos rígidos, boot pela unidade C e com acesso ao drive de CD-ROM da máquina hospedeira.

```
# vi vmWindows.sh

#!/bin/bash

kvm -cdrom /dev/cdrom -m 512 -hda \ /home/aluno/maquinas_virtuais/
windows.img -hdb \ /home/aluno/maquinas_virtuais/disco2.img -boot c
```

## Interfaces de rede

O KVM cria uma rede virtual dentro do processo que contém a máquina virtual. Essa rede é composta por um servidor DHCP, um servidor DNS e um gateway. Por padrão, a interface `eth0` vai funcionar atrás de um NAT, e o processo do KVM vai tunelar todo o tráfego de rede em espaço de usuário. É uma solução muito prática e funciona muito bem para soluções simples, entretanto, o desempenho fica limitado. Para soluções mais complexas, onde queremos permitir o acesso externo a uma máquina virtual, é necessário mapear uma interface TAP no host com uma interface no KVM.

A configuração de rede no KVM é feita pelo parâmetro `-net`, as principais:

- `-net user[,option]`: é o modo default do KVM, não precisa de privilégios de administrador e configura a interface em NAT para acesso a internet. Possui várias opções de parâmetro, onde é possível especificar o endereço MAC da interface, servidor de DNS, servidor Wins, servidor DHCP, entre outros;
- `-net nic[,options]`: cria uma nova interface de rede e conecta na VLAN `n` (por padrão `n=0`). É possível definir qual o modelo da interface (`model=`), endereço MAC, endereço IP, entre outros. Para listar os modelos de interface disponíveis, execute o comando  
`# kvm -netnic,model=?`
- `-net tap[,options]`: esse modo permite que o `qemu` crie uma bridge ligando a interface da máquina virtual à interface `tap` do host. Dessa forma, a máquina virtual será acessível pela rede externa.
- `-net none`: desativa as interfaces de rede na máquina virtual. Somente a interface de loopback é ativada.

Quando omitimos o parâmetro `-net`, o KVM utiliza o modo `user` como padrão.

Antes de especificar qual será o driver utilizado pela sua interface de rede virtual, é recomendado verificar quais os dispositivos disponíveis. O comando `# kvm -netnic,model=?` listará os drivers de rede suportados pelo KVM.

```
$ kvm -net nic,model=?  
  
qemu: Supported NIC models: ne2k_pci,i82551,i82557b,i82559er,rtl8139,  
e1000,pcnet,virtio
```

Podemos ainda definir qual será o endereço MAC da interface de rede utilizando a opção `macaddr`. No exemplo a seguir, iniciaremos a máquina virtual com o endereço MAC `52:54:00:12:34:56`.

```
# kvm -m 512 -hdalinux.img -net nic,model=rtl8139,macadr=  
52:54:00:12:34:56 -net user
```

Para adicionar novas interfaces de rede, basta incluir mais parâmetros `-net` no comando de execução da máquina virtual. Exemplo de uma máquina virtual com duas interfaces de rede:

```
# kvm -net nic,model=rtl8139 -net nic,model=rtl8139 -m 512 -hda  
linux.img
```

Quando iniciamos uma máquina virtual, ela não está acessível por nenhuma outra máquina da rede, inclusive o próprio host. Como alternativa, podemos criar um redirecionamento de porta `tcp` para permitir conexões SSH entre o host e o guest.

```
# kvm -m 512 -hdalinux.img -net nic -net user,hostfwd=tcp::5555-:22
```

Ao iniciarmos a máquina virtual com a opção `hostfwd=tcp::5555:22`, criamos um redirecionamento da porta TCP 5555 do host para a porta 22 do guest. Com isso, conseguimos acessar o guest com o comando `ssh`.

No host execute:

```
# ssh -p 5555 aluno@localhost
```

## Redes virtuais

- Quando uma máquina virtual é iniciada no KVM, ela fica isolada, ou seja, sem acesso ao host ou a qualquer outra máquina da rede.
- Para permitir a comunicação entre a máquina virtual e os outros equipamentos da infraestrutura, devemos criar uma rede virtual.



Para que uma máquina virtual possa ser incluída em uma rede virtual, ela precisa ser iniciada com pelo menos uma interface de rede do tipo tap, pois é esse tipo de interface que permite o mapeamento da interface da máquina virtual no host.

A criação das redes virtuais é feita através de interfaces de rede do tipo bridge criadas no host. Essas interfaces funcionarão como switches de camada 2, direcionando o tráfego na camada na enlace e, dessa forma, permitirão que as máquinas virtuais se comuniquem com outras máquinas e com a internet.

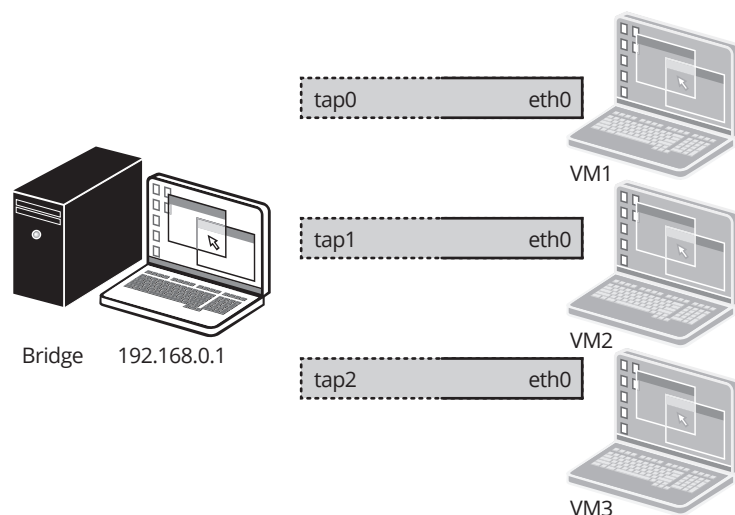
### Configurando uma rede privada entre máquinas virtuais

As interfaces do tipo bridge são criadas com comando `brctl`, que está disponível no pacote `bridge-utils`. Cada rede virtual é representada por uma bridge, e cada interface das máquinas virtuais é mapeada em uma interface tapN.

Vamos imaginar o cenário ilustrado na Figura 1.7: um host com três máquinas virtuais, onde cada máquina foi iniciada com uma interface de rede do tipo tap e o host tem uma bridge, chamada br0, configurada com o endereço IP 192.168.0.1.

Para fazer com que todas as máquinas virtuais se comuniquem, precisamos de duas etapas:

1. Configurar as interfaces eth0 das máquinas virtuais com um IP da rede 192.168.0.0/24.
2. Adicionar as interfaces tapN na bridge configurada no host.



**Figura 1.7**  
Host configurado com uma bridge e três máquinas virtuais.

1. Criando a bridge no host:

```
# brctladdbr br0
```

2. Configurando o endereço IP:

```
# ifconfig br0 192.168.0.1
```

3. Adicionando as interfaces TAP das máquinas virtuais:

```
# brctladdrif br0 tap0  
# brctladdrif br0 tap1  
...
```

4. Configurando o endereço IP:

```
VM1 - # ifconfig eth0 192.168.0.2 netmask 255.255.255.0  
VM2 - # ifconfig eth0 192.168.0.3 netmask 255.255.255.0
```

Para testar o funcionamento, execute o comando *ping* de uma máquina virtual para outra.

## Acessando a rede do host

Existem duas formas de permitir que uma máquina virtual acesse a rede a qual o host está conectado: através de NAT ou através do acesso direto.

Para permitir que as máquinas virtuais acessem a rede fazendo NAT, precisamos habilitar o encaminhamento de pacotes e configurar o iptables do host.

1. Ativar o encaminhamento de pacotes e configurar o iptables:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

2. Configurar o iptables para realizar o encaminhamento dos pacotes:

```
# iptables -P FORWARD ACCEPT  
# iptables -A POSTROUTING -t nat -s $VM_NETWORK -o $INTERNET_NIC -j  
MASQUERADE  
onde:  
$VM_NETWORK = 192.168.0.0/24  
$INTERNET_NIC = eth1
```

A outra forma de permitir que uma máquina virtual acesse a rede do host é adicionar uma interface física do host na bridge que corresponde à rede das máquinas. Dessa forma, a interface do host adicionada encaminha os pacotes originados pelas máquinas virtuais para a rede do host.

## Automatizando a configuração de redes virtuais

É possível criar um script para automatizar a inclusão de novas interfaces tap na bridge. O arquivo */etc/kvm/kvm-ifup* é executado sempre que o KVM inicia uma máquina virtual. Podemos modificá-lo ou criar um novo arquivo para realizar inclusão das máquinas na rede virtual.

O opção `script` do parâmetro `-net` permite especificar o caminho do script que será executado na inicialização da máquina virtual.

Exemplo do script `/etc/kvm/kvm-ifup` modificado:

```
#!/bin/sh

set -x

switch=br0

if [ -n "$1" ];then

    /sbin/ip link set $1 up

    sleep 0.5s

    /usr/sbin/brcctladdif $switch $1

exit 0

else

echo "Error: no interface specified"

exit 1

fi
```

Iniciando a máquina virtual com um script diferente do `/etc/kvm/kvm-ifup`:

```
# kvm -m 512 -hdaimagem.img -net nic -net tap,script=/var/adm/
scripts/kvm-tap.sh
```

## Snapshots

O termo "snapshot" teve origem no universo da fotografia, onde era utilizado para designar fotos instantâneas. Atualmente, é um termo muito usando em computação para designar uma cópia de um sistema, filesystem, memória, banco de dados, entre outros em um determinado momento.

Diferente de um backup convencional, onde todos os dados são copiados em um intervalo longo de tempo, um snapshot cria uma cópia do estado atual em alguns segundos. Isso ocorre porque os snapshots trabalham apenas com ponteiros e praticamente não fazem a movimentação dos dados.



O tamanho de um snapshot está relacionado à modificação dos ponteiros dos arquivos, e não ao seu tamanho real.

O KVM permite a criação de snapshots dos discos das máquinas virtuais, entretanto, apenas o formato `qcow2` tem suporte a essa tecnologia.

## Gerenciando snapshots

- Criando snapshots com o KVM:

```
#qemu-img snapshot -c snapshot1 \ /home/aluno/virtual_machines/  
windows.img
```

- Listando os snapshots de uma imagem:

```
# qemu-img snapshot -l /home/aluno/virtual_machines/windows.img
```

- Restaurando um snapshot:

```
# qemu-img snapshot -a snapshot1 \ /home/aluno/virtual_machines/  
windows.img
```

- Excluindo um snapshot:

```
# qemu-img snapshot -d snapshot1 \ /home/aluno/virtual_machines/  
windows.img
```





# Roteiro de Atividades 1

## Atividade 1.1 – Gerenciando snapshots

---

Nesta atividade, vamos explorar as funcionalidades dos snapshots. De acordo com o material apresentado, desenvolva as seguintes atividades:

1. Após a instalação, desligue a máquina virtual Linux e tire um snapshot com o nome *instalacao* (sem cedilha ou til).
2. Inicie a máquina virtual Linux e crie o arquivo */home/aluno/processos.txt* com a lista dos processos.
3. Desligue a máquina e restaure o snapshot *instalacao*.
4. Reinicie a máquina virtual e verifique o que aconteceu com o arquivo */home/aluno/processos.txt*.

## Atividade 1.2 – Gerenciando a interface de rede

---

Nesta atividade, utilizaremos os conceitos vistos na parte teórica sobre como manipular as interfaces de rede pela linha de comando do KVM.

1. Inicie a máquina virtual *linux.img* com duas interfaces de rede (uma do tipo *nic*) e defina o modelo da interface (um como *virtio* e outro como do tipo *tap*).



Não se esqueça de verificar se a sua versão do KVM possui o driver virtual para esse tipo de interface de rede.

2. Altere o endereço MAC das interfaces de rede.

Você pode usar a linha de comando abaixo para gerar um endereço MAC aleatório:

```
# printf '%02x:' $(( 0x$(od /dev/urandom -N1 -t x1 -An | \
cut -c 2-) & 0xFE | 0x02)); od /dev/urandom -N5 -t x1 -An \
| cut -c 2- | sed 's/ /:/g'
```

3. Configure a interface virtual *eth0:0* do host com o endereço IP 192.168.X.10/24 e configure a interface *eth0* da máquina virtual com o endereço IP 192.168.X.20/24. Teste a comunicação entre o host e a máquina virtual com o comando *ping*. Obteve sucesso? Por quê?

- 
4. Crie uma bridge com o nome de *lab* e a configure com o endereço IP 192.168.X.1.
  5. Adicione as interfaces *tap*, referentes à máquina virtual, à bridge *lab*.
  6. Adicione uma rota para que todo o tráfego da rede 192.168.X.0/24 passe pela bridge *lab*.
  7. Teste a comunicação entre o host e a máquina virtual.





### Atividade 1.3 – Convertendo máquinas virtuais

---

1. Converta a imagem da máquina virtual, criada no VirtualBox, que está no diretório informado pelo instrutor para o formato *qcow2* do KVM.
2. Inicie a máquina virtual e teste seu funcionamento.

# 2

## Administração de máquinas virtuais com KVM

### objetivos

Aprender o gerenciamento de máquinas virtuais, instalação de dispositivos, filtragem de pacotes e configuração de interfaces.

Gerenciamento de máquinas virtuais, Virt-Manager, bridges, libvirt, hardware da máquina virtual e migração de máquinas virtuais.

### conceitos

### Virtualização no Linux

- Atualmente existem diversos softwares que permitem a criação de máquinas virtuais.
- Os mais conhecidos no Linux são VMWare, VirtualBox e KVM.



O KVM é um ótimo virtualizador gratuito disponível para inúmeros tipos de uso, inclusive em ambientes corporativos com alta demanda de processamento e disponibilidade. O KVM é um subsistema do Linux que, quando utilizado com o Qemu, permite a criação de máquinas virtuais que executam com baixo overhead, isto é, o Sistema Operacional na máquina virtual funciona em uma velocidade próxima da real (caso estivesse na máquina física).

### Ferramentas para gerenciamento de máquinas virtuais

A administração do ambiente virtualizado é simplificado quando apoiado por ferramentas gráficas que auxiliam o administrador de sistemas no gerenciamento das máquinas virtuais. No caso do KVM, destacamos as ferramentas de gerenciamento a seguir, que auxiliam o gerenciamento.

- QEMU-KVM;
- Eucalipytus;
- OpenNebula;
- Nimbus;
- Ovirt;
- Plataforma em nuvem Xen;
- OpenQRM;
- Virt-Manager.



## QEMU-KVM

Com esse pacote é possível executar a virtualização no Linux utilizando somente a linha de comando. Para tanto, além de instalar o QEMU-KVM, é necessário recompilar o kernel do Sistema Operacional com a flag `CONFIG_KVM` habilitada.



O QEMU-KVM deve ser obtido no site do KVM e compilado seguindo sua documentação. Também é necessário instalar o módulo *kvm.ko*, que é gerado quando a flag `CONFIG_KVM` é habilitada, conforme o exemplo:

```
# modprobe kvm
```

Quando o módulo KVM é instalado, deve-se verificar se não há erro no `dmesg`, uma vez que o módulo pode ser carregado e, mesmo assim, apresentar algum problema.

O próximo passo é criar a imagem de disco. Para tanto, é necessário utilizar a ferramenta `qemu-img` e passar às opções desejadas, como, por exemplo, o tamanho da imagem, o formato e o nome:

```
# qemu-img create <imagem><tamanho>
```

Note que podemos criar outros tipos de imagem, tais como *qcow*, *qcow2*, *cow*, *vdi*, *vmdk*, entre outras, passando a opção `-f`.

Uma vez carregado esse módulo, é necessário passar o parâmetro `-enable-kvm` para o QEMU, para que este consiga gerenciar a máquina virtual de maneira eficiente, tirando proveito das características de emulação dos processadores modernos, denominadas VT-d e VT-X. Uma alternativa é a utilização do comando *qemu-kvm*, que já assume essa opção, conforme o exemplo:

```
# qemu-kvm -hda <imagem> -m <quantidade de memoria em MB>
```

A mensagem “open /dev/kvm: No such file or directory” é um sinal de que o módulo KVM não foi habilitado corretamente. Em alguns casos, para o módulo ser carregado corretamente, deve-se habilitar a opção de virtualização na BIOS do computador em questão.

Caso seja necessário passar alguma imagem adicional, basta utilizar o parâmetro `-hd` acompanhado de uma letra, como `-hdb`, `-hdc` etc. Caso queira tornar a imagem um CD-ROM, basta utilizar a opção `-cdrom`. Exemplo prático:

Crie um disco rígido virtual de 8 GB:

```
#qemu-img create -f raw debian-kde.raw 8G
```

■ **-f raw**: indica que o formato da imagem é *raw*. Os principais formatos são *raw* e *qcow2*.

Instale um sistema no HD virtual criado com o comando anterior.

```
#kvm -m 512-name maquina1 -drive file=/caminho/imagem.iso,media=cdrom -drive file=/caminho/debian-kde.raw,media=cdrom,cache=writeback
```

■ **-m 512**: indica que a VM terá 512 MB de memória RAM;

■ **-drive file=**: indica o caminho e o nome do arquivo de imagem de CD, e `media=cdrom` indica que essa imagem será a de um CD-ROM. A outra opção existente para “media” é `media=disk`, para indicar que é um disco rígido.

A novidade no comando anterior é `cache=writeback`, usado para melhorar a performance do disco. O padrão é `cache=writethrough`, que é mais seguro, porém possui desempenho pior, principalmente quando usado com imagens no formato *qcow2*.

Os Sistemas Operacionais mantêm um cache de página na memória RAM para melhorar o desempenho de I/O de disco. As operações de gravação são consideradas concluídas após os dados terem sido copiados para o cache de página. As operações de leitura são satisfeitas a partir do cache de página, se os dados solicitados estiverem no cache. No ambiente do KVM, tanto o Sistema Operacional do host quanto o do guest podem manter seus próprios caches de páginas, resultando em duas cópias de dados na memória. Em geral, é melhor ignorar pelo menos um desses caches de página. As opções de cache são: `writethrough`, `writeback`, `unsafe` e `none`.

- **cache=none:** como o nome já sugere, desabilita o cache para leitura e gravação no host, transformando todas as requisições de I/O do guest em operações Direct I/O no host.

Para os outros três modos, o cache é ativado tanto para leitura quanto para gravação, porém o cache de gravação é tratado de forma diferente em cada caso.

- **cache=writethrough:** fará com que o cache de página do host seja usado para ler e escrever dados, mas a confirmação de escrita será enviada para o guest somente quando os dados forem relatados como escrito pelo sistema de armazenamento.
- **cache=writeback:** também usará o cache para ler e escrever, mas relatará os dados escritos como concluídos, assim que os dados estiverem presentes no cache de página do host. Este é seguro, desde que o host também seja. Se o host desligar por falta de energia, pode ocorrer corrupção de dados no guest.
- **cache=unsafe:** usará o cache para ler e escrever e não será necessário gravar os dados no disco, podendo mantê-los em cache. Se alguma coisa der errado, como o desligamento anormal do host, a imagem de disco da VM poderá ser inutilizada.

## Eucalyptus

Um dos pacotes de software livre mais populares para a construção de infraestruturas de computação em nuvem é o Eucalyptus (sigla que significa Elastic Utility Computing Architecture for Linking Your Programs to Useful Systems). O que faz dele único é o fato de sua interface ser compatível com a Amazon Elastic Compute Cloud (Amazon EC2 — a interface de computação em nuvem da Amazon). Além disso, o Eucalyptus inclui o Walrus, que é um aplicativo de armazenamento em nuvem compatível com o Amazon Simple Storage Service (Amazon S3 — a interface de armazenamento em nuvem da Amazon).

O Eucalyptus suporta KVM/Linux e Xen para hypervisors, e inclui a distribuição de cluster Rocks para o gerenciamento de cluster.

## OpenNebula

O OpenNebula é outro aplicativo de software livre interessante (sob licença da Apache), desenvolvido na Universidad Complutense de Madrid. Além de dar suporte à construção em nuvem privada, o OpenNebula apoia a ideia de nuvens híbridas. Uma nuvem híbrida permite combinar uma infraestrutura em nuvem privada com uma infraestrutura em nuvem pública (como a da Amazon), para permitir graus ainda maiores de escalada. O OpenNebula suporta Xen, KVM/Linux e VMware, e é amparado por elementos como libvirt, para o gerenciamento e introspecção.

## Nimbus

O Nimbus é outra solução IaaS com foco na computação científica. Com o Nimbus, podemos arrendar recursos remotos (como os fornecidos pela Amazon EC2) e gerenciá-los localmente (configurar, implementar VMs, monitorar etc.). O Nimbus tem origem no projeto Workspace Service (parte do Globus.org). Por depender da Amazon EC2, o Nimbus suporta Xen e KVM/Linux.

## oVirt

O pacote oVirt é uma ferramenta de gerenciamento de VM aberta que aumenta de um pequeno número de VMs para milhares de VMs sendo executadas em centenas de hosts. O pacote oVirt, desenvolvido pela Red Hat, é um console de gerenciamento baseado na web que, além do gerenciamento tradicional, suporta a automação do gerenciamento em cluster e do balanceamento de carga. A ferramenta oVirt foi composta na linguagem Python.

## Plataforma em nuvem Xen

A Citrix integrou a Xen em uma plataforma IaaS, usando a Xen como um hypervisor ao incorporar outros recursos de software livre como o Open vSwitch. Uma vantagem interessante da solução Xen é o foco no gerenciamento baseado em padrões – incluindo OVF, Distributed Management Task Force (DMTF), Common Information Model (CIM) e a Virtualization Management Initiative (VMAN) – do projeto Kensho. A pilha de gerenciamento Xen suporta garantias SLA em conjunto com métricas detalhadas para estorno.

## OpenQRM

É caracterizada como uma plataforma de gerenciamento de datacenter. A OpenQRM fornece console único para gerenciar um datacenter virtualizado completo que é plugável, de maneira arquitetônica, para permitir a integração de ferramentas de terceiros. A OpenQRM integra suporte para alta disponibilidade (através de redundância) e suporta uma variedade de hypervisors, incluindo KVM/Linux, Xen, VMware e Linux VServer.

## Virt-Manager

O sistema de virtualização do KVM conta com um sistema de gerenciamento chamado Virt-Manager, que provê ambiente gráfico através do qual se pode criar, configurar e alterar os parâmetros da máquina virtual a ser usada, incluindo memória, configurações de rede e disco. Há também uma série de utilitários adicionais para efetuar algumas tarefas auxiliares:

- **virt-install**: criação de máquinas virtuais novas;
- **virt-clone**: criação de uma cópia de uma máquina virtual existente;
- **virt-img**: criação de uma nova imagem de disco, a ser usado para simular um disco virtual;
- **virt-viewer**: possibilita a interação com a interface gráfica da máquina virtual.

## Instalação do gerenciador gráfico Virt-Manager

A execução do pacote GDM instalará vários outros pacotes para tornar possível o login gráfico. Após a instalação, faça login no host e abra no menu principal do Virt-Manager.

Para usar o Virt-Manager, é preciso se certificar de que as camadas subjacentes estejam devidamente iniciadas:

```
# apt-get install gdm
# modprobe kvm-{intel,amd} //(escolha apenas um)//
# modprobe tun //(a configuração de rede virtual depende disto)//
# /etc/init.d/libvirt start
```

O ambiente gráfico do Virt-Manager pode ser iniciado por meio do comando *virt-manager*, utilizando as credenciais do administrador do sistema.

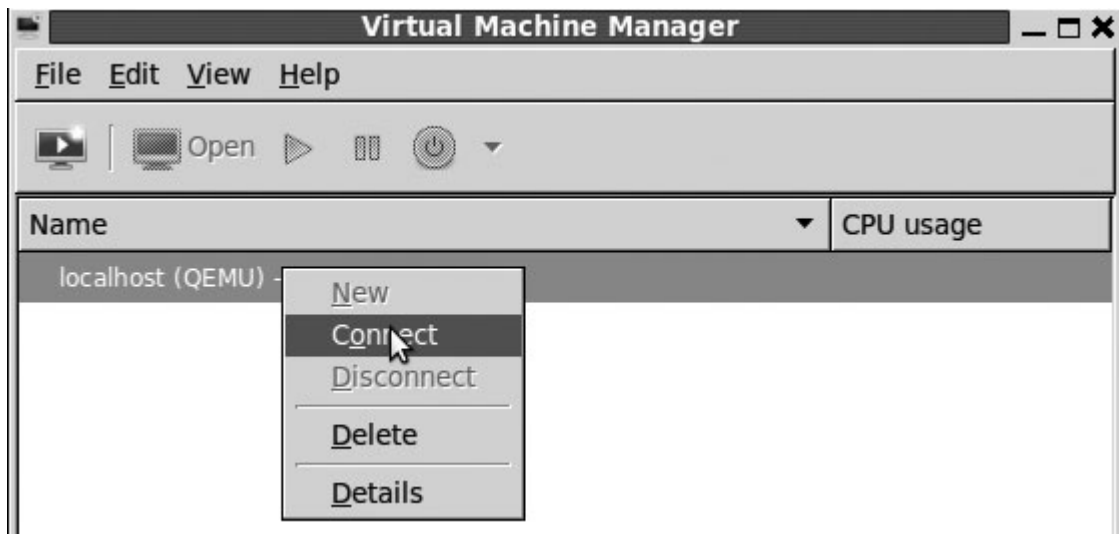
## Exercício de fixação 1

### Instalando o pacote Virt-Manager

Instale e inicie o pacote Virt-Manager pelo repositório do Debian.

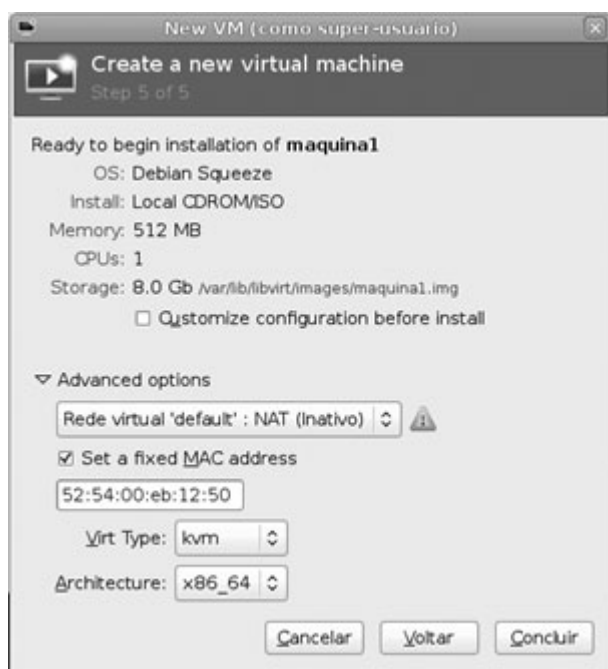
## Criando uma máquina virtual

Para criar uma máquina virtual no Virt-Manager, é necessário se conectar ao host.



**Figura 2.1**  
Conexão do  
gerenciador de  
VMs.

Com a conexão realizada, a criação da máquina virtual seguirá alguns passos básicos para definição de informações como, por exemplo, o nome da máquina virtual, a forma de instalação, tipo de Sistema Operacional e sua versão, quantidade de memória máxima e espaço máximo de hard disk que a máquina virtual utilizará. Ao final do processo de criação, o sistema vai apresentar uma tela com as informações gerais da máquina virtual.



**Figura 2.2**  
Configuração da máquina virtual.

Ao concluir a instalação, a máquina virtual fará a instalação do Sistema Operacional.

## Acessando a máquina virtual

Para acessar a máquina virtual a partir do Virt-Manager, basta clicar duas vezes sobre a máquina virtual na aba principal e, então, abrirá uma nova janela que mostrará o conteúdo da tela da máquina virtual, tanto o modo console quanto a interface gráfica (quando apropriado).

Basta clicar com o botão direito sobre o nome da máquina virtual para mudar seu estado. Por exemplo, para iniciá-la.

Também é possível acessar a máquina virtual através de uma conexão segura, como é o caso do SSH, ou mesmo por meio do protocolo **VNC**. Para acessá-la via VNC, é necessário instalar um servidor VNC na máquina virtual, iniciar o daemon vncd e então conectar-se a ela com um cliente VNC, como o vncviewer.

### VNC

Virtual Network Computing é um protocolo construído para possibilitar interfaces gráficas remotas. Através desse protocolo, um usuário pode conectar-se a um computador a distância e usar as suas funcionalidades visuais como se estivesse sentado em frente ao computador.

## Configurando a interface de rede da máquina virtual

O KVM permite criar redes virtuais, recurso muito interessante para o administrador de sistemas, pois permite criar redes isoladas dentro do mesmo ambiente virtualizado, viabilizando virtualizar todos os serviços internos e externos da rede.

A interface gráfica do Virt-Manager permite efetuar configurações em todos os componentes da máquina virtual de forma bastante simplificada. Durante o processo de criação da interface de rede da máquina virtual, é possível definir em qual rede virtual a interface de rede dessa máquina virtual estará conectada.

Para configurar a interface de rede utilizando a interface gráfica Virt-Manager, devemos selecionar a máquina virtual e acessar o menu Host Details e, na aba de “Redes Virtuais”, clicar em “+”, para adicionar rede. É preciso inserir algumas informações, como o nome da rede virtual, endereço IP intervalo para alocação do servidor DHCP e qual rede física a rede virtual deverá estar conectada para comunicação com os clientes externos.

Após criar a interface de rede virtual, adicione à máquina virtual usando a opção “I”. Depois clique em “Show Virtual Machine Details”/“Add Hardware”, e então selecione “Network”, escolha o Host Device previamente criado, e finalmente clique em “Avançar” e “Concluir”.

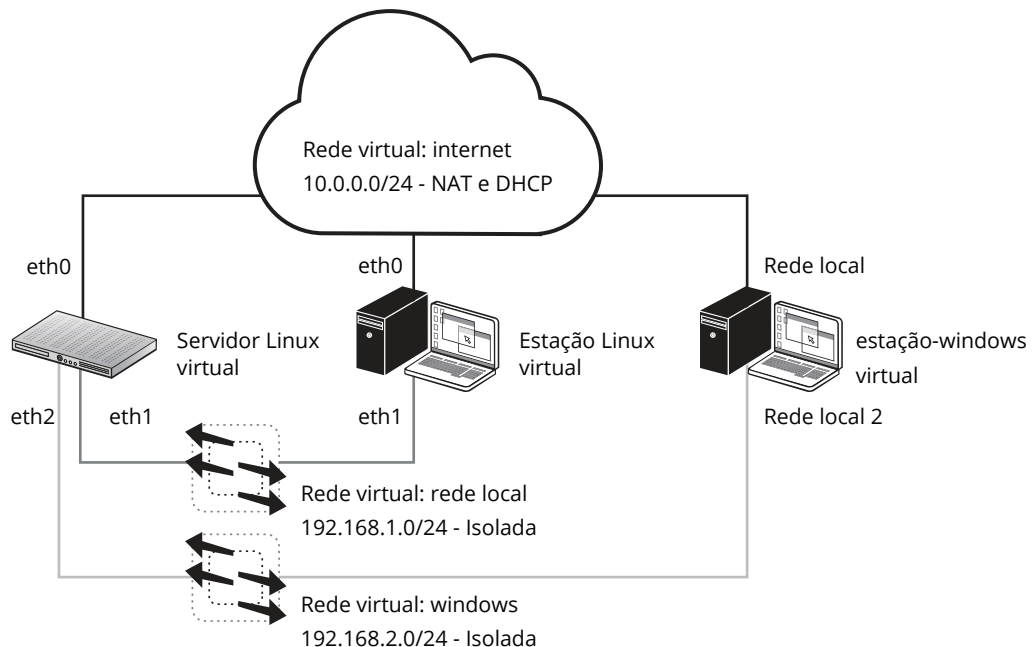
A nova interface de rede aparecerá na lista de dispositivos da máquina virtual e poderá ser utilizada assim que a máquina virtual for inicializada.

## Exercício de fixação 2

### Criando uma rede virtual

- Crie uma rede virtual com o nome “internet”, ative o DHCP, configure a faixa dos IPs que serão distribuídos para 10.0.0.128 até 10.0.0.254, e escolha NAT como tipo de conexão.
- Crie uma rede virtual chamada “rede-local”. Não ative o DHCP: configure o endereço da rede para 192.168.1.0/24 e marque que será uma rede virtual isolada.
- Crie uma rede virtual chamada “windows”. Não ative o DHCP: configure o endereço da rede para 192.168.2.0/24 e marque que será uma rede virtual isolada.

Nosso objetivo é montar uma estrutura semelhante à da imagem a seguir:



**Figura 2.3**  
Estrutura da rede virtual.

## Exercício de fixação 3

### Criando uma máquina virtual

Crie uma máquina virtual utilizando a imagem do Windows fornecida pelo instrutor e inclua a máquina na rede virtual “windows”.

## Exercício de fixação 4

### Importando uma máquina virtual

Crie uma nova máquina virtual chamada “servidor” e importe a imagem do disco virtual criada no primeiro Capítulo. Use a opção “Import existing disk image”. Inclua essa máquina virtual na redes “rede-local” e “internet”.



Verifique no arquivo `/etc/libvirt/quem/nome_da_máquina.xml` se todas as informações de configuração da máquina virtual estão corretas.



Algumas versões do Virt-Manager apresentam um bug referente à importação do tipo de imagem do disco, sempre assumindo que a imagem do disco importado é do tipo RAW. Pode ser necessário alterar para o tipo correto.

## libvirt

A libvirt se encarrega de criar as interfaces virtuais e associá-las à bridge que desejarmos. Basta informarmos, na configuração das máquinas virtuais, o nome do dispositivo de bridge:

Sempre que possível, utilizar também o virtio é uma boa opção, pois esse driver reduz o overhead decorrente da virtualização de I/O. Trata-se de um driver paravirtualizado.

## Filtragem de pacotes

É possível filtrar pacotes usando o netfilter/iptables na sua bridge. Nela, os pacotes destinados à máquina física usarão a cadeia INPUT, e os que forem destinados às máquinas virtuais passarão pela cadeia FORWARD.



Como o filtro para a máquina local será aplicado à interface do bridge, é necessário utilizar um módulo do netfilter específico para esse caso: o módulo `physdev` (de physical device, dispositivo físico).

Dois exemplos de regras para negar conexões à porta TCP 23 na interface `br0`, recebidas pela interface física `eth1` ou saindo da porta UDP 9876 pela interface física `eth0`:

```
# iptables -A INPUT -m physdev --device-in eth1 -p tcp --dport 23 -j DROP

# iptables -A INPUT -m physdev --device-out eth0 -p udp --sport 9876 -j DROP
```

Além disso, esse módulo também é necessário para aplicar regras de filtragem às cadeias de repassagem de pacotes FORWARD e POSTROUTING. Por exemplo, para negar na bridge a repassagem de pacotes destinados ao IP 10.1.2.5:

```
# iptables -A FORWARD -m physdev --device-is-bridged -d 10.1.2.5 -j DROP
```

Alguns outros procedimentos ainda mais específicos, como a completa desativação do netfilter em dispositivos de bridge, são cobertos na documentação da Red Hat.

Se desejamos filtrar pacotes IP já na camada Ethernet, claramente violando o modelo TCP/IP – mas com benefícios potenciais de desempenho –, talvez valha a pena experimentar o `ebtables` (de Ethernet Bridge Tables). Trata-se de um filtro de frames Ethernet capaz de atuar também na camada IP. O `ebtables` é independente do netfilter/iptables, mas pode atuar em conjunto com este para oferecer maior flexibilidade.

## Hardware da máquina virtual

- Na sua essência, o “hardware” da máquina virtual é idêntico ao hardware da máquina real, fazendo com que o Sistema Operacional na máquina virtual pareça estar executando diretamente sobre um computador real.



- Cada máquina virtual possui todas as características inerentes a um PC completo, incluindo BIOS e configuração do Setup.



A máquina virtual é formada pelo seu próprio “hardware” virtual. Ela tem seu processador, memória, placa de rede, disco rígido e até mesmo pode usufruir de dispositivos como USB e portas paralelas e seriais, de maneira independente da máquina física. É natural que a capacidade de processamento seja uma fatia do processador físico. O mesmo acontece com a memória e outros componentes.

Podemos definir uma máquina virtual como uma nova máquina (como se tivesse sido comprada), mas que existe dentro de outra máquina física. Uma máquina física pode ser fatiada em várias máquinas virtuais, cada uma pode conter um Sistema Operacional diferente, com programas diferentes. Não existe compartilhamento de arquivos ou partes de Sistemas Operacionais entre as máquinas virtuais – o isolamento entre elas é total.

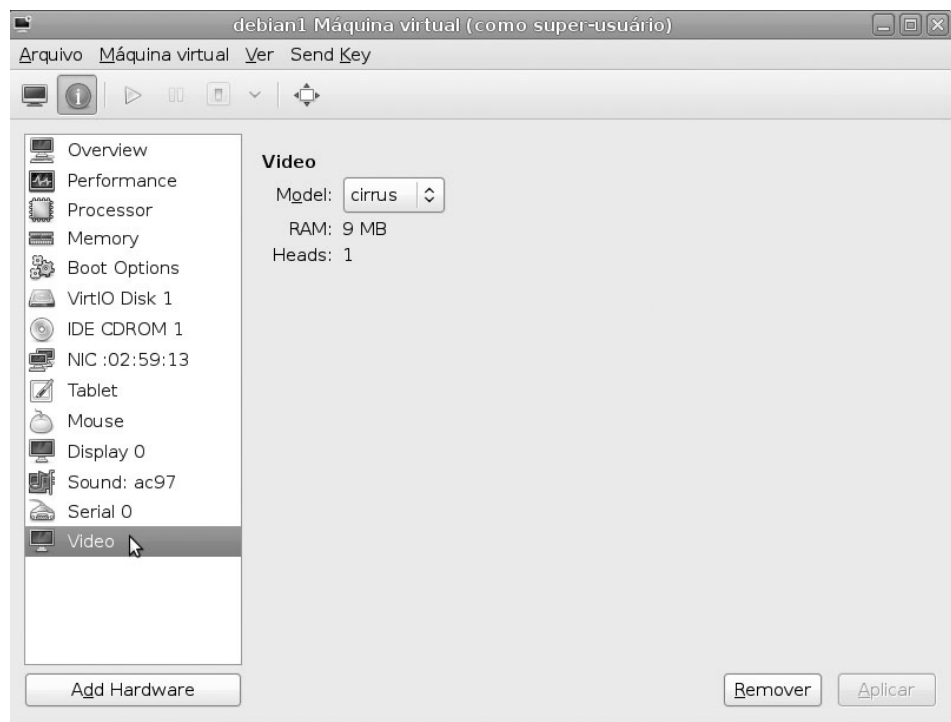
As placas de rede da máquina virtual podem ser conectadas a switches virtuais, que darão acesso à rede física ou não. A partir do momento em que criamos uma máquina virtual e definimos seu tamanho, arquitetura de discos e rede, basta instalar o Sistema Operacional escolhido e aplicativos desejados. Ao final desse processo a máquina virtual estará pronta e será idêntica a uma máquina física, que poderá compartilhar normalmente arquivos com outra máquina virtual ou física através da rede.

## Placa de vídeo

Para adicionar uma placa gráfica Cirrus ao sistema virtual utilizando interface texto, basta executar o comando:

```
#qemu-kvm -m 1024 -name f15 -drive file=/images/f15.img,if=virtio  
-vga cirrus
```

Para adicionar via interface gráfica.

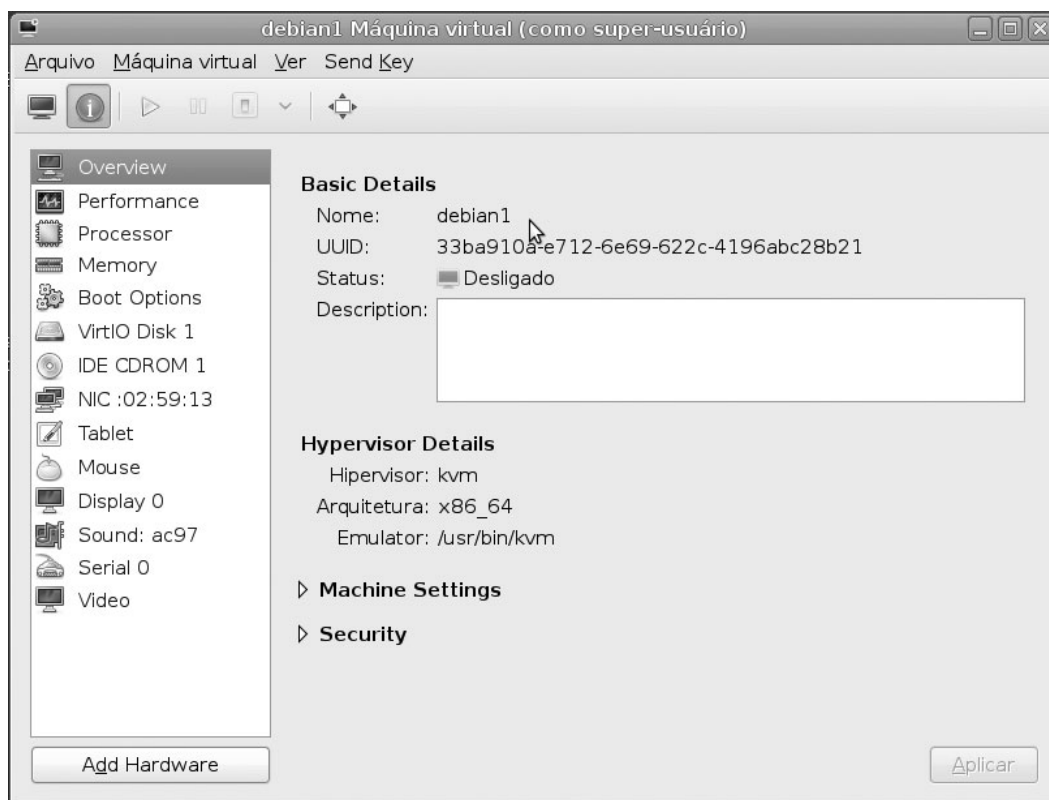


**Figura 2.4**  
Interface gráfica de  
configuração da MV  
(placa de vídeo).



## Detalhes da máquina virtual

- Gerenciador de Performance.
- Detalhes do Processador.
- Alocação de Memória.
- Opções de inicialização do sistema.
- Detalhes do Armazenamento.
- Dispositivo de Áudio.



### Gerenciador de Performance

Na opção “Performance” (desempenho), serão apresentados quatro gráficos sobre utilização do sistema que permitem monitorar a utilização dos recursos do hardware:

- CPU;
- Memória;
- Disk I/O;
- Network I/O.

### Detalhes do Processador

Nesse item serão apresentados detalhes sobre o processador da máquina virtual, com o número de processadores alocados e o máximo que pode ser alocado pela máquina virtual.

### Alocação de Memória

Nesse item é apresentada a alocação de memória da máquina virtual, contendo informações sobre a alocação atual. Nesse módulo é possível alterar a quantidade de memória alocada, em MB, pela máquina virtual.

**Figura 2.5**

Interface gráfica de configuração da MV (overview).

## Opções de inicialização do sistema

Podemos definir o dispositivo que vai inicializar o sistema virtual por meio do comando:

```
# qemu-kvm -drive file=/images/imagel.raw,index=0,media=disk
```

Utilizando a interface gráfica, podemos também definir o device que inicializará a máquina virtual. E ainda podemos definir se a máquina virtual será inicializada em tempo de boot do host hospedeiro.

## Detalhes do armazenamento

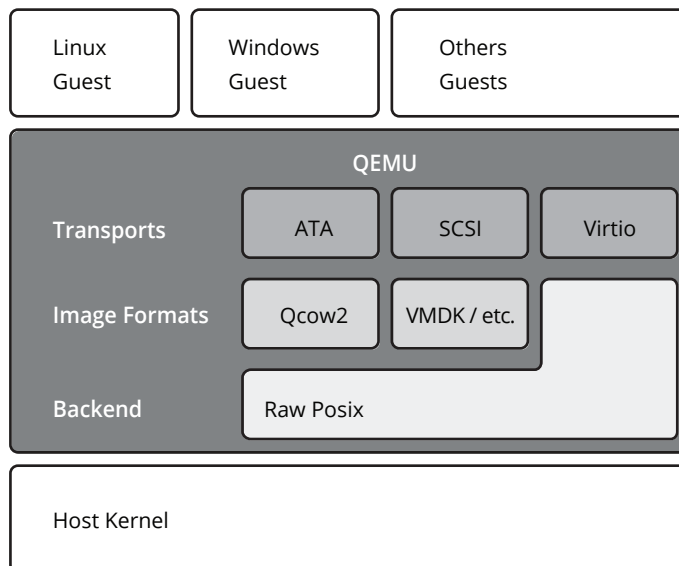
Nesse item será detalhado o disco virtual alocado na máquina virtual, detalhando o Dispositivo de Destino, que se refere às informações do dispositivo na máquina virtual, e o Caminho de Origem, referente às informações do dispositivo na máquina hospedeira.

## Dispositivo de Áudio

Apresenta os detalhes do dispositivo de áudio configurado na máquina virtual.

## Adicionando novo storage à máquina virtual

O QEMU fornece um simples controlador ATA/IDE por padrão e funciona com quase todos os Sistemas Operacionais. Alternativamente, o QEMU pode emular um controlador SCSI e fornece os dispositivos virtualizados utilizando o framework VirtIO.



**Figura 2.6**  
Estrutura do QEMU.

Exemplos de comandos para adicionar um novo storage à máquina virtual, por meio de linha de comando:

```
#kvm -drive file=/dev/volumes/maquina01,media=disk,index=0,boot=on,i  
f=virtio  
  
#kvm -drive file=/dev/volumes/maquina01,media=disk,index=0,boot=on,i  
f=ide  
  
#kvm -drive file=/dev/volumes/maquina01,media=disk,index=0,boot=on,i  
f=scsi
```

Utilizando a interface gráfica, devemos fazer da seguinte forma:



## Migração de máquinas virtuais

Com o ambiente já virtualizado, podem existir picos de uso da máquina virtualizada, algumas vezes por falhas no escalonamento do projeto de virtualização, de forma que a máquina física não suporte toda a carga e cause lentidão. Ao mesmo tempo, é possível que existam outras máquinas físicas disponíveis e ociosas. Assim, para melhor aproveitamento da capacidade dos servidores, os operadores de TI podem efetuar migrações de máquinas virtualizadas para outras máquinas físicas, melhorando a velocidade de todos os processos e aproveitando ao máximo os benefícios da virtualização. A migração pode ser feita de três formas:

- **Static Migration:** quando a máquina virtual é desligada usando os métodos de seu Sistema Operacional e logo em seguida é movida para outra máquina física;
- **Cold Migration:** quando a máquina virtual é suspensa (exemplo: o Hibernar do sistema Windows), usando métodos de seu Sistema Operacional ou acionada a função de Snapshot, que salva o estado atual da máquina virtualizada. Após acionado um dos dois procedimentos citados, a máquina virtualizada é movida para outra máquina física e iniciado o sistema virtualizado em seu estado anterior;
- **Live Migration:** quando o próprio monitor de máquinas virtuais, que executa a máquina virtualizada, se incumba de fazer a migração sem paradas. Live Migration permite o equilíbrio de carga dinâmica de associações virtualizadas do recurso, a manutenção do hardware sem tempo ocioso da máquina virtualizada e o apoio dinâmico contra falhas do sistema. Veja a seguir como fazer o Live Migration.

No host A, devemos iniciar o KVM normalmente:

```
#kvm -drive file=/caminho/imagem-vm.img, ...
```

No host B, iniciar o KVM usando o mesmo comando usado para iniciar o host A, porém deve ser incluído o parâmetro `-incoming`.

```
#kvm -drive file=/caminho/imagem-vm.img, ... -incoming tcp:0:4444
```

No monitor da máquina virtual em execução no host A, execute o comando `migrate`:

```
(qemu) migrate -d tcp:host_b:4444
(qemu) info migrate
Migration status: active
```

**Figura 2.7**  
Interface gráfica  
para adicionar novo  
storage.

```
transferred ram: 17343 kbytes  
remaining ram: 1023952 kbytes  
total ram: 1057216 kbytes
```

Ao final do comando, a máquina virtual continuará sua execução no host B.

Cuidados que devem ser observados:

- As imagens ou dispositivos de discos virtuais devem estar acessíveis no host de destino;
- Interfaces tap devem estar na mesma rede local;
- As versões do KVM devem ser exatamente a mesma;
- Migrate -b envia os dispositivos de bloco juntos (block migration).

O maior desafio da migração de máquinas virtuais é fazer com que a máquina virtualizada se comporte da mesma maneira, após a migração entre os servidores físicos.

Com o gerenciador Virt-Manager, é possível executar o processo de migração com apenas alguns cliques, mas para que essa migração ocorra, são necessários alguns requisitos independentemente da ferramenta de gerenciamento utilizada.

- É obrigatório que o disco virtualizado esteja acessível entre a origem e destino – exemplo: Network File System (NFS) armazenado em pasta compartilhada;
- As máquinas de origem e destino devem estar acessíveis via rede;
- As plataformas dos Sistemas Operacionais de origem e destino devem ser compatíveis;
- No caso de processadores com tecnologia de virtualização sendo usada pela máquina virtual, esta não poderá ser migrada para outro servidor que não disponha dessa tecnologia.





# Roteiro de Atividades 2

## Atividade 2.1 – Live Migration

---

Live Migration é um recurso que permite que uma máquina virtual migre de um host para outro sem interromper o seu funcionamento. Para realizar a migração, são necessárias algumas características no ambiente virtual, tais como:

- Ter no mínimo dois hosts;
- Os hosts devem ter o mesmo storage compartilhado;
- Os hosts devem possuir a mesma configuração de rede virtual.

Esta atividade deverá ser realizada em dupla. Chamaremos as máquinas dos alunos de host1 e host2. O host1 deverá estar configurado com o endereço IP 10.0.X.1/24 e o host2, com o endereço 10.0.X.2/24.

### Preparando o ambiente virtual

#### **Passo1: Montando o filesystem do storage compartilhado.**

Para que possamos realizar o Live Migration, necessitamos que todas as máquinas virtuais do nosso ambiente sejam criadas em uma área compartilhada por todos os hosts.

Utilizaremos o NFS (que será visto em detalhes no Capítulo 3) para realizar o compartilhamento dos dados entre os hosts do nosso ambiente virtual.

As máquinas host1 e host2 deverão montar o filesystem compartilhado pelo servidor do instrutor.

### Montar o filesystem

Criar o ponto de montagem:

```
# mkdir /storagekvm
```

Montar o filesystem do servidor:

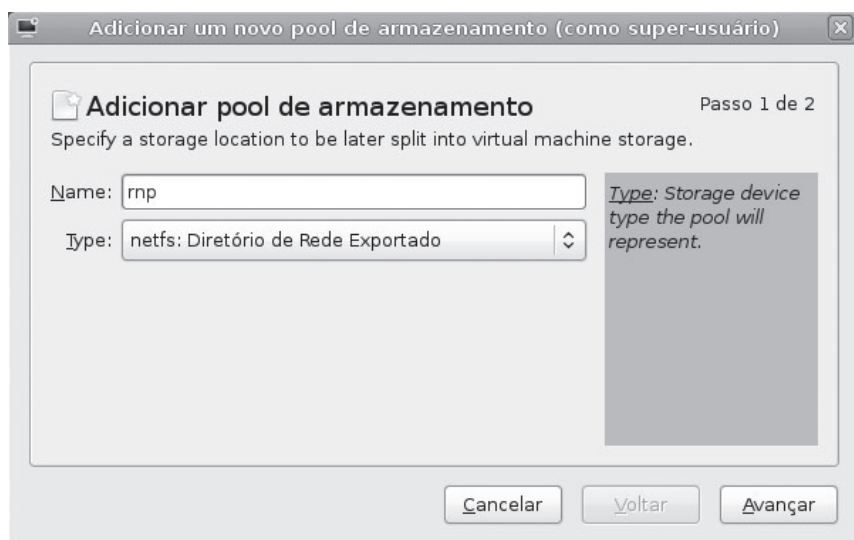
```
# mount -t nfs <IP_SERVIDOR_SALA_DE_AULA>:/storagekvm/duplaX /  
storagekvm
```

### Configuração dos hosts

1. Abra o Virt-Manager e acesse o menu “Editar”, clique em “Host Details” e em seguida clique sobre a aba “Storage”.
2. Clique sobre o botão com um sinal de “+”, localizado no canto inferior esquerdo.
3. Defina o nome do storage como “rnp” e selecione o tipo “netfs: Diretório de Rede Exportado”. Clique em “Avançar”.

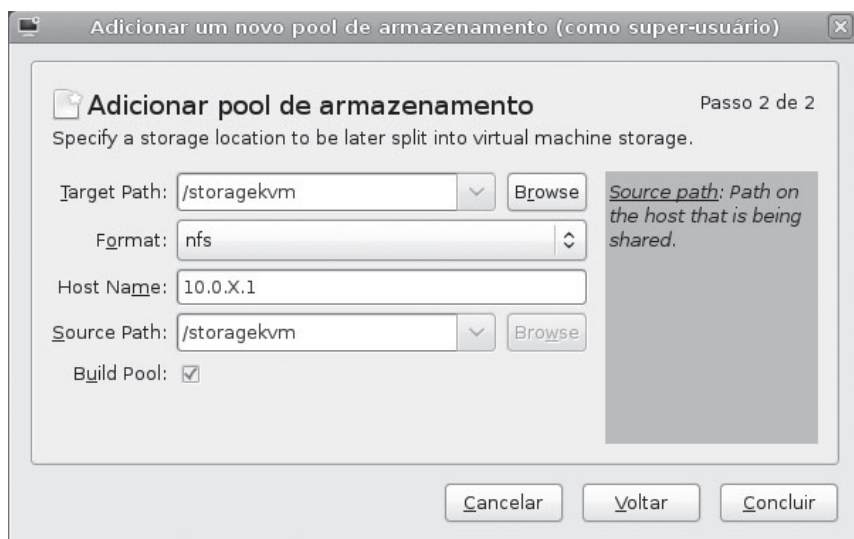






4. Configure com as seguintes opções:

- Target Path: /storagekvm;
- Format: nfs;
- Host Name: 10.0.X.1;
- Source Path: /storagekvm/duplaX.



■ **Passo 2: Criando uma máquina virtual no storage rnp.**

1. Crie uma nova máquina virtual utilizando o arquivo `/storagekvm/machineX.img` como disco virtual.

Na interface do Virt-Manager, clique sobre o nome do host (exemplo: localhost) com o botão direito e escolha a opção “Novo”.

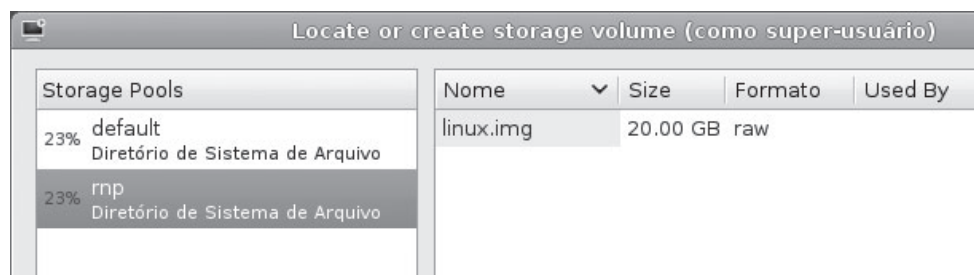
- Determine o nome da máquina virtual como "migration", marque a opção "Import existing disk image" e clique em "Avançar".



- Informe o tipo de Sistema Operacional, a versão e em seguida clique sobre o botão "Browse..." para localizar o arquivo de imagem de disco.



- Será exibida a janela de Storage Pools do Virt-Manager. Clique sobre o storage "rnp" e em seguida clique sobre o arquivo de imagem de disco "linux.img".



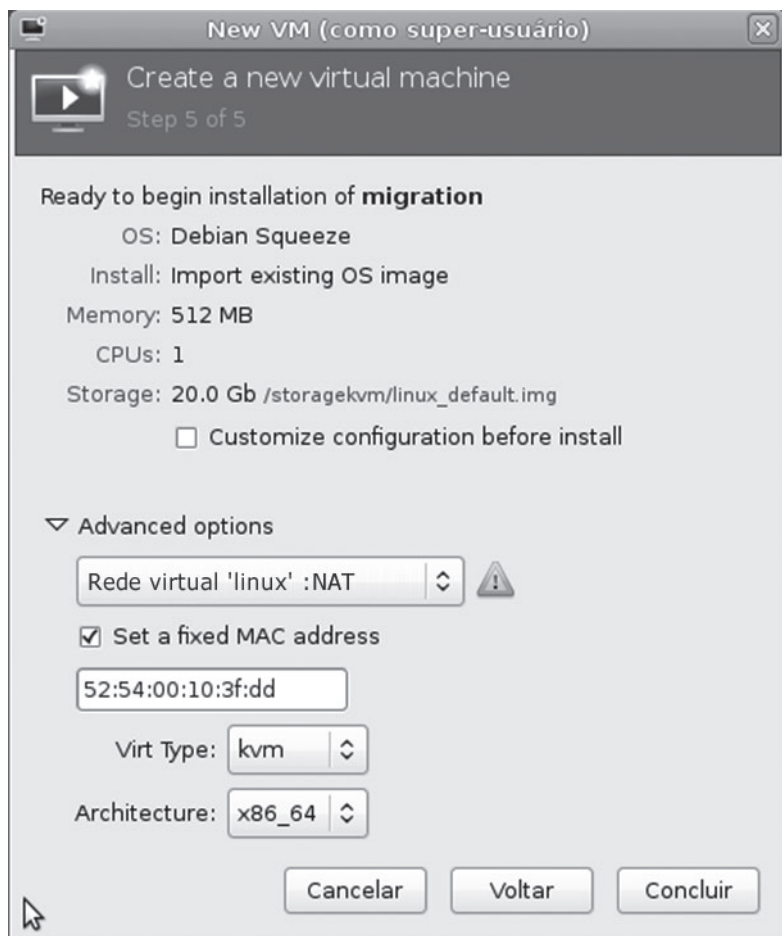
5. Após selecionar o arquivo que será o disco da máquina virtual, clique em “Avançar”.



6. Informe a quantidade de memória e o número de processadores que serão utilizados pela máquina virtual:

- Memória: 512 MB;
- Processadores: 1.

7. Em “Advanced Options”, especifique a rede virtual “linux” e clique em “Concluir”.



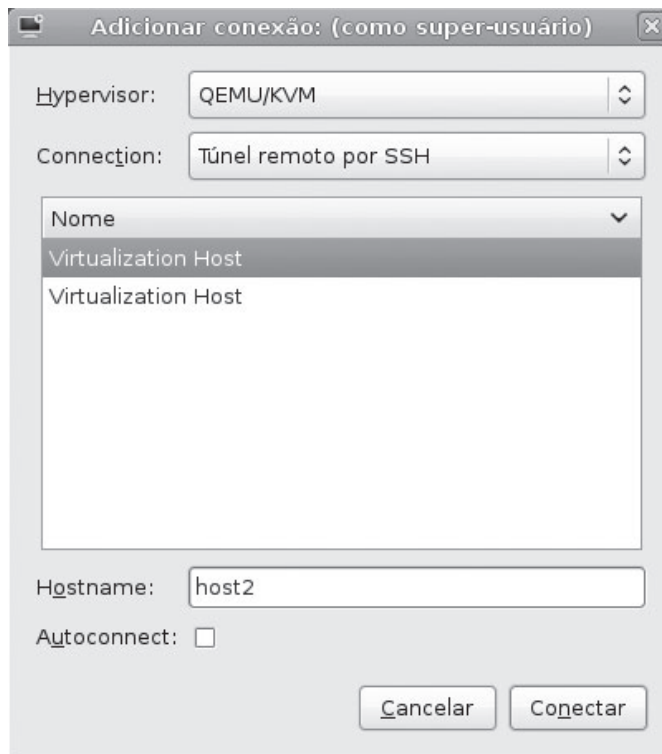
8. Inicie a máquina virtual no host1.

### Passo 3: Adicionando novos hosts ao ambiente virtual.

Adicionando o host2 ao Virt-Manager do host1.

1. Abra o Virt-Manager e clique sobre o menu “Arquivo”. Em seguida, sobre “Add Connection...” e selecione:

- Hypervisor: QEMU/KVM;
- Connection: Túnel remoto por SSH.



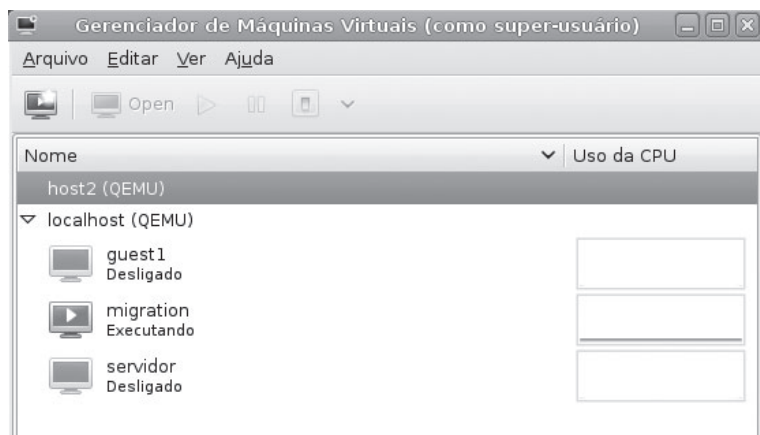
Como não estamos utilizando um servidor de DNS, será necessário informar o nome dos hosts do ambiente virtual no arquivo `/etc/hosts`. Exemplo:

```
10.0.X.1 host1
10.0.X.2 host2
```



Para que os hosts se conectem automaticamente, deve-se configurar o acesso SSH sem senha.

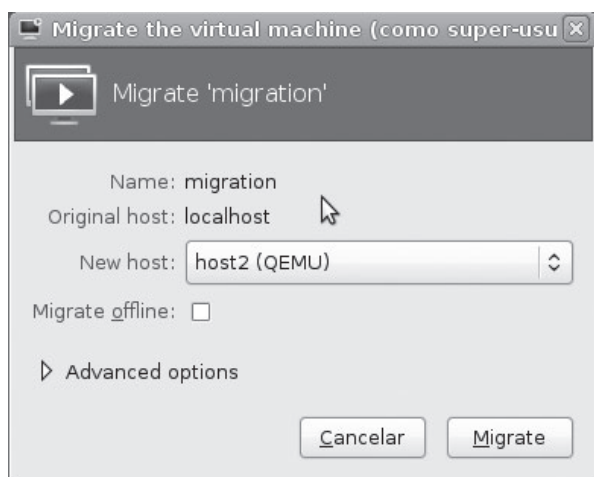
2. Agora nosso ambiente virtual possui dois hosts conectados.



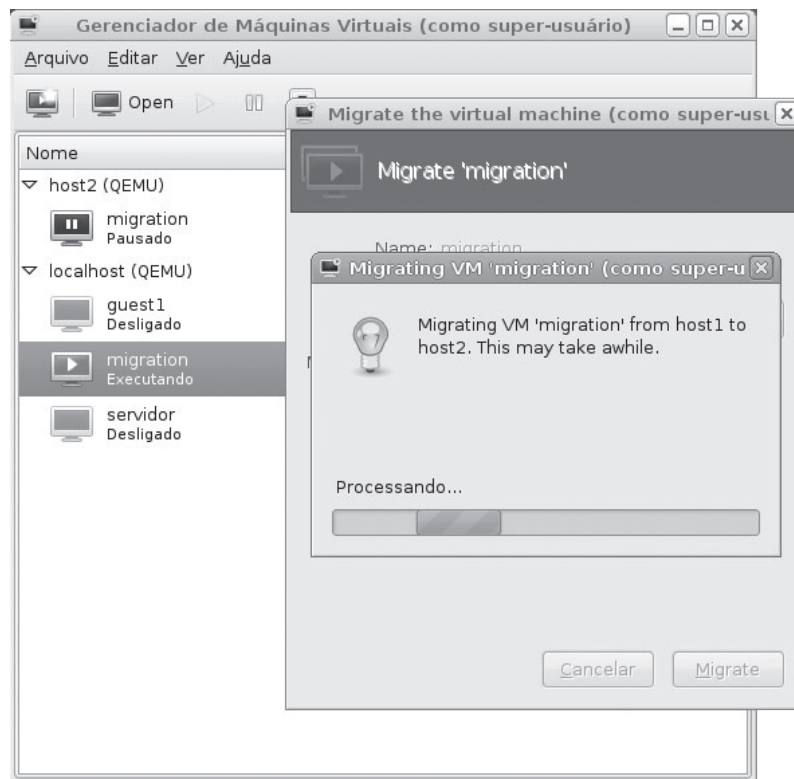
#### Passo 4: Migrando uma máquina virtual entre os hosts.

Agora que nosso ambiente está todo preparado, podemos iniciar o processo de Live Migration da nossa máquina virtual.

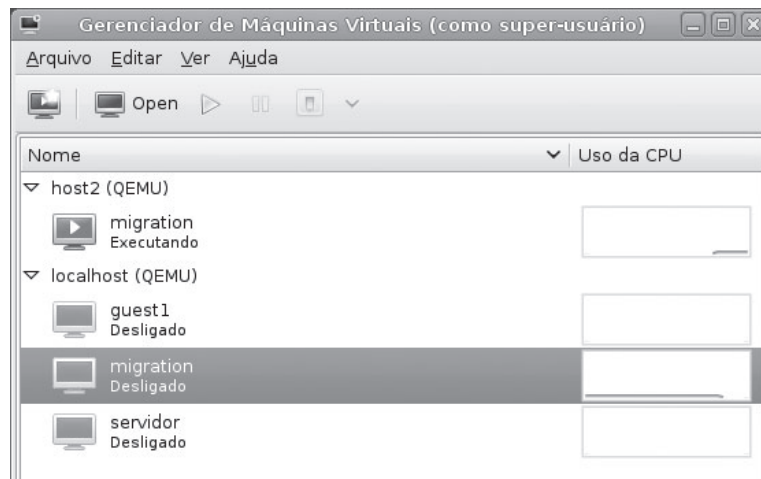
1. Acesse o console da máquina virtual “migration” e execute o comando top, para que os processos que estão rodando na máquina fiquem aparecendo.
2. No host1, com o botão direito do mouse, clique sobre o nome da máquina virtual “migration” e em seguida clique em “Migrate”.
3. Em “New host”, selecione o host de destino (host2) e clique sobre o botão “Migrate”.



4. Clique em “Advanced options”, marque a opção “Address”, informe o endereço IP do host de destino e clique sobre o botão “Migrate”.
5. Aguarde o processo de migração da máquina virtual.



6. Ao final da migração, a máquina virtual aparecerá ligada no host2. Acesse o console dessa máquina e verifique se o comando `top` ainda está rodando.



## Atividade 2.2 – Criando clones

---

Para dar continuidade às atividades do curso, em muitos momentos precisaremos de duas máquinas virtuais Linux. A geração de clones de máquinas virtuais é um recurso que permite otimizar a criação de novas máquinas.

Utilizando esse recurso, crie um clone da máquina virtual servidor, chamado cliente-linux.

# 3

## DNS e NFS

### objetivos

Conhecer os sistemas Domain Name Service (DNS) e Network File System (NFS).

Domain Name Service, sistema de tradução de endereços, Network File System, sistema de arquivos em rede.

### conceitos

### Introdução

O estudo deste capítulo se inicia com os conceitos do serviço de resolução de nomes (DNS), abordando sua utilização e execução de maneira prática e intuitiva, passando pela instalação, configuração e testes. Na segunda parte serão apresentados os conceitos e utilização do Network File System (NFS).

Na primeira parte são apresentados os conceitos e definições comuns ao serviço DNS, tais como:

- Espaço de Nomes (DNS);
- Domínio;
- Zonas de autoridade;
- Registro de recursos;
- Mapeamento direto e reverso.

Em seguida, serão vistos os tipos e montagens típicas de um serviço DNS, que envolvem:

- Servidor primário;
- Servidor(es) Secundário(s);
- Servidor Cache;
- Servidores raiz da internet.

Na etapa seguinte são abordados os aspectos do DNS associados à sua utilização no Linux, desde instalação e configuração, até formas de teste e diagnóstico.

Por último vêm as atividades práticas:

- Instalação do servidor DNS;
- Configuração do servidor DNS Primário;
- Configuração do servidor DNS Secundário.





# Domain Name Service (DNS)

O que é o DNS?

- Serviço para tradução de nomes em endereços IP e vice-versa.
- Também denominado Domain Name System ou Domain Name Server.



O Serviço de Tradução de Nomes, ou simplesmente serviço de nomes, é um componente dos serviços em redes que permite a aplicativos e serviços localizar e fazer referência a informações em um ambiente de rede. Os nomes são parte crítica dos serviços em redes, pois ajudam na administração de todos os recursos disponíveis na rede.

Domain Name Service (DNS) fornece serviços de nomes e diretórios em redes que implementam a pilha de protocolos Transmission Control Protocol/Internet Protocol (TCP/IP). Podemos considerá-lo como um sistema de banco de dados distribuído, que traduz nomes de estações em endereços, fornecendo informações sobre roteamento e domínios.

O DNS é essencial para os sites que se conectam diretamente à internet. Porém, mesmo para redes locais isoladas que utilizam protocolos da pilha TCP/IP, sempre faz sentido utilizá-lo. É particularmente importante para o sistema de correio eletrônico, pois é nele que são definidos registros que identificam a máquina que manipula as correspondências relativas a um dado nome, identificando, assim, onde determinado usuário recebe suas correspondências.

## Por que utilizar DNS?

Primeiro motivo:

- Máquinas possuem maior facilidade em lidar com números.
- Pessoas preferem usar nomes.
- Exemplo:
  - Página da RNP.
  - Mais fácil usar www.rnp.br que http://200.130.77.75



Segundo motivo:

- Torna o acesso aos serviços de rede independente do endereço IP do servidor.
- Troca do IP do servidor é transparente para o usuário, pois a localização do IP é fornecida pelo serviço DNS.

O esquema de endereçamento utilizado pela internet baseado em endereços IPv4 (números de 32 bits utilizados especificamente pela camada de rede) não é recomendado para uso em navegadores e outros serviços. Desse modo, um usuário da internet, ao usar seu navegador, pode digitar um endereço que lhe é familiar, como www.rnp.br, em vez de algo como 200.130.77.75, possibilitando, dessa forma, que empresas e pessoas possam criar endereços baseados em nomes, associando-os ao nome da empresa ou produto.

Cabe ao serviço DNS fazer a associação entre o nome escolhido pelo usuário e o endereço IPv4 de 32 bits.

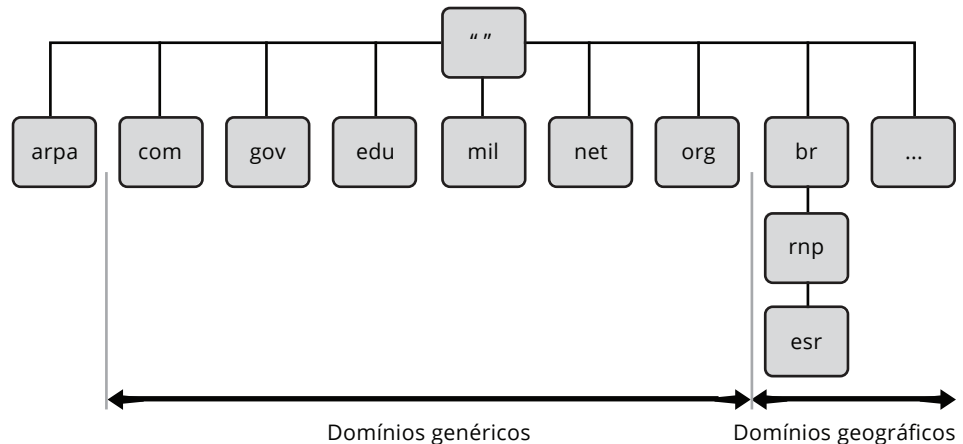
Frequentemente o administrador de rede tem de mudar a faixa de endereço utilizada pelos vários servidores que controla. Tal alteração pode ser necessária pela mudança do bloco de endereços disponibilizado, podendo ocorrer em função de uma mudança de provedor de acesso à internet, o que provoca a alteração de endereços para o novo bloco de endereços disponibilizado pelo novo provedor.



Um motivo adicional para o uso de um sistema de endereçamento baseado em nomes é torná-lo independente do endereço IP em uso em determinado momento.

Podem ainda ocorrer modificações em função da realocação de endereços do provedor ou mesmo da própria rede do usuário, por causa de rearranjos causados por ampliações ou modificações na topologia da rede.

Desse modo, um usuário que acessa o serviço via internet, ou mesmo via intranet, não precisa se preocupar com alterações na topologia e/ou endereçamento do servidor, já que estará acessando o servidor pelo nome, cabendo ao DNS manter essa associação atualizada.



**Figura 3.1**  
Espaço de nomes  
do DNS.

O DNS é um dos componentes básicos da internet. Podemos considerá-lo como um sistema de banco de dados distribuídos de nível mundial, que traduz endereços simbólicos de estações em endereços IP numéricos e vice-versa.

Quando se usa serviços não transparentes como Telnet, FTP, SSH e SFTP, o usuário pode fornecer como argumento um endereço simbólico de estação (o nome de um computador hospedeiro). Funções resolvidoras (resolver functions) do DNS traduzem esse endereço simbólico para um endereço IP numérico.

O rápido crescimento da internet levou à introdução dos conceitos de domínios e subdomínios DNS. Os domínios compõem a estrutura básica do DNS e podem ser comparados a diretórios em um sistema de arquivos, com cada domínio aparecendo como subárvores de informações no espaço de nomes de domínio. Um endereço simbólico de estação em um domínio deve ser único, porém esse mesmo endereço simbólico pode existir em outros domínios. Os domínios são implementados como uma hierarquia, que pode ser vista como uma árvore invertida, começando no domínio-raiz, cujo endereço simbólico real é "", ou seja, um rótulo (label) nulo.

O domínio raiz é implementado por um grupo de servidores de nomes chamados de servidores raiz. Os servidores raiz só têm informações completas a respeito dos domínios de topo, que se localizam imediatamente abaixo do domínio-raiz na árvore de hierarquia de domínios. Os domínios de topo, por sua vez, têm apontadores para os servidores em domínios de níveis inferiores. Nenhum servidor, nem mesmo os servidores raiz, tem informações completas sobre todos os domínios, mas os seus apontadores podem indicar que outros servidores podem fornecer a informação desejada. Assim, os servidores raiz podem até não saber a resposta para uma pergunta, mas certamente saberão para quem direcioná-la.

## Definições

- Domínio.
- Zonas de autoridade.
- Registro de recursos.
- Mapeamento:
  - ▣ Direto.
  - ▣ Reverso.



A segurança da informação é um ponto crítico para a sobrevivência das organizações na era da informação. Vários são os problemas envolvidos, no entanto a sociedade depende das informações armazenadas nos sistemas computacionais para tomadas de decisão em negócios, uso de órgãos do governo, entre outros contextos.

A informação pode existir em diversos formatos: impressa, armazenada eletronicamente, transmitida pelo correio convencional ou eletrônico etc. Seja qual for o formato, meio de armazenamento ou transmissão, recomenda-se que ela seja protegida adequadamente.

Sendo assim, é de responsabilidade da segurança da informação protegê-la de vários tipos de ameaças, a fim de garantir a continuidade do negócio, minimizar riscos e maximizar o retorno sobre os investimentos. Felizmente, é crescente a conscientização das organizações frente ao valor e as vulnerabilidades de seus ativos, no que diz respeito à segurança.

Hoje, a segurança da informação é determinante para assegurar competitividade, lucratividade, atendimento aos requisitos legais e a boa imagem da organização junto ao mercado e às organizações, tanto no setor público quanto no setor privado. Em tais contextos, a segurança da informação é um componente que viabiliza negócios como o e-Gov (governo eletrônico) ou e-commerce (comércio eletrônico).

## Domínio

- Subárvore do espaço de nomes DNS.
- Nome de domínio completo, também denominado Fully Qualified Domain Name (FQDN), consiste em:
  - ▣ Nome da máquina.
  - ▣ Nome do domínio.
  - ▣ Domínio de topo.
  - ▣ Exemplo:
    - ▣ www.rnp.br
    - ▣ www: máquina (ou host).
    - ▣ rnp: domínio.
    - ▣ br: domínio de topo (top level domain).



Se precisássemos lembrar todos os endereços IP das páginas da web que visitamos diariamente, ficaríamos loucos. Seres humanos não são bons em lembrar séries de números. No entanto, somos bons na lembrança de palavras, e por isso usamos os nomes de domínios. Você possui, provavelmente, vários nomes de domínios guardados em sua cabeça. Exemplos: rnp.br, receita.fazenda.gov.br, g1.com.br, mit.edu, google.com etc.

As partes “com”, “edu” e “br” desses servidores são chamadas de domínios principais ou domínios de primeiro nível. Existem vários domínios principais, incluindo “com”, “edu”, “gov”, “mil”, “net” e “org”, além das siglas de duas letras de cada país, para identificar a origem, em inglês. Não custa lembrar que os domínios registrados nos EUA não possuem a identificação (“us”). Em cada domínio principal existe uma enorme lista de domínios secundários. Novos tipos de domínio foram criados para oferecer melhor identificação. Exemplos mais conhecidos são: professor – “prof.br”, pessoais – “nom.br”, rede de televisão – “tv.br” etc.

Um domínio é uma subárvore do espaço de nomes DNS. Um domínio completo, também denominado de Fully Qualified Domain Name (FQDN), consiste basicamente em um nome de máquina, um nome de domínio e um domínio de topo. O endereço www.rnp.br é um exemplo de FQDN, onde:

- **www**: nome da máquina (ou host);
- **rnp**: nome do domínio;
- **br**: domínio de topo.

Cada subárvore é considerada parte de um domínio. Nesse caso, “rnp” faz parte do domínio “br”. Outra situação é referente a subdomínios dentro da própria rede. Considere o endereço www.df.rnp.br, no qual “df” é uma subárvore de “rnp” e faz parte desse domínio. A seguir, alguns exemplos.

Domínio rnp:

- **www.rnp.br**: máquina www no domínio rnp.br.
- **mail.rnp.br**: máquina mail no domínio rnp.br.

Domínio df.rnp.br:

- **www.df.rnp.br**: máquina www no domínio df.rnp.br.
- **mail.df.rnp.br**: máquina mail no domínio df.rnp.br.

## Zonas de autoridade

- Também denominadas Start of Authority (SOA).
- Domínios e zonas de autoridade não são sempre sinônimos:
  - ▣ Um nome de domínio se refere a um único ponto no espaço de nomes.
  - ▣ Uma zona de autoridade refere-se ao local no qual estão armazenados os dados sobre as máquinas (hosts) do domínio.



Cada nome de domínio possui um registro de zona de autoridade que apresenta informações do domínio e da zona em que o domínio está inserido. Entre essas informações, estão:

- Nome do servidor de nomes primário;
- Endereço eletrônico (e-mail) do responsável pelo domínio;
- Número de série (serial number) da zona;
- Intervalo de refresh: indica o tempo, em segundos, que o servidor de nomes secundário deve verificar por atualizações no servidor de nomes primário;
- Intervalo de retry: indica o tempo, em segundos, que o servidor de nomes secundário aguarda por uma resposta do servidor de nomes primário, antes de indicar uma falha;



- Tempo de expire: informa o tempo, em segundos, após o qual o servidor não mais responde por informações daquela zona;
- Tempo minimum: informa o Time To Live (TTL) default caso o domínio não especifique um TTL;
- Tempo de vida, Time To Live (TTL): valor passado pelo servidor de nomes indicando, para a máquina que originou a pergunta, o tempo que a informação pode ser mantida em cache.

## Exercício de fixação 1

### DNS authoritative answer

Qual das alternativas abaixo representa uma resposta non-authoritative answer quando realizamos uma consulta a um servidor de DNS?

- (1) Indica que o tempo de vida (TTL) expirou.
- (2) Representa uma resposta originada do cache local.
- (3) Indica um erro no registro SOA.
- (4) Retorna a informação do próximo servidor DNS.
- (5) Indica que o cache do servidor de DNS está envenenado (DNS Poisoning).

## Registro de recursos

- Todos os domínios podem ter um conjunto de registro de recursos associado.
- A verdadeira função do DNS é mapear nomes de domínio em registros de recurso.



Um registro de recurso é composto por cinco campos. Embora eles sejam codificados em binário para maior eficiência, a maioria dos registros de recurso é apresentada como texto de ASCII, sendo uma linha por registro de recurso, como é mostrado a seguir: Domínio, Tempo de vida (TTL), Classe, Tipo e Valor.

O campo *Domínio* refere-se ao domínio para o qual esse registro é aplicado. Normalmente, existem muitos registros para cada domínio e cada cópia do banco de dados carrega informação sobre múltiplos domínios. Esse campo é a chave primária de procura usada para satisfazer as buscas. A ordem dos registros no banco de dados não é significativa. Quando uma busca é feita sobre um determinado domínio, são devolvidos todos os registros pedidos, emparelhados em classe.

O campo *Tempo de vida (Time To Live)* dá uma indicação do nível de estabilidade do registro. Informações muito estáveis recebem valor alto, como 86400 (o número de segundos em um dia). Informações altamente voláteis têm um valor pequeno, como 60 (um minuto).

O terceiro campo de todo registro de recurso é a *Classe*. Para informação de servidor internet, é sempre IN. Para informações não relacionadas com a internet, são usados outros códigos, que na prática são vistos raramente.

O campo *Tipo* fornece a informação do tipo de registro. Os tipos mais importantes são:

- **Start of Authority Information (SOA):** contém informações referentes ao servidor de nomes DNS do domínio, versão do banco de dados DNS, e-mail do administrador responsável pelo domínio etc;
- **A (Host Address):** mantém a tabela de endereços IP dos hosts mantendo compatibilidade com a tabela antiga de hosts. Permite mapear um nome de host para um ou cada endereço IP;

- **Name Server Identification (NS):** especifica os servidores de nomes responsáveis pelo domínio, zona ou subzona;
- **General Purpose Pointer (PTR):** permite obter um nome de host, a partir do conhecimento do seu endereço IP. É a contraparte do registro A;
- **Canonical Name Alias (CNAME):** permite criar um apelido para um host. Esse registro é útil para ocultar detalhes de implementação da sua intranet, por exemplo: ftp.marketing.corporação.com pode ser apenas um apelido do verdadeiro servidor que executa ftp do marketing;
- **Host Information (HINFO):** permite identificar propriedades de hardware e do Sistema Operacional do host que serão exibidas toda vez que o usuário acessar esse host. A padronização de identificação do tipo de CPU e do Sistema Operacional deve obedecer aos nomes listados na RFC 1700;
- **MAIL Exchange (MX):** mantém informações referentes aos hosts responsáveis pelo e-mail do domínio.

Finalmente, nós temos o campo *Valor*. Esse campo pode ser um número, um nome de domínio, ou um conjunto de caracteres ASCII. A semântica depende do tipo de registro.

Considere que todos os domínios podem ter associado um conjunto de registro de recursos. Conforme mencionado anteriormente, uma zona de autoridade refere-se a um local onde estão armazenados os dados sobre as máquinas do domínio.

Todos os registros de um determinado domínio estão em uma zona de autoridade responsável por este. Nesse caso podemos ter, considerando os domínios df.rnp.br e rnp.br, duas zonas de autoridade. A primeira responsável pelas informações referentes aos servidores daquele domínio, e a segunda, responsável pelas informações dos servidores do outro domínio. Nesse caso, podemos ter:

- **Servidor 1:** SOA do domínio rnp.br e responsável pelas informações de DNS dessa rede;
- **Servidor 2:** SOA do domínio df.rnp.br e responsável pelas informações de DNS dessa rede.

Esses servidores podem estar inclusive em localidades diferentes, sem relação direta (primário e secundário) um com o outro.

Os registros do banco de dados do DNS possuem o seguinte formato:

```
<nome> <TTL> <classe> <tipo> <dados>
```

- Nome: identifica o objeto. Exemplo: um computador.
- TTL: tempo que o registro deve ser mantido em cache.
- Classe: define o tipo de servidor.
  - Servidor Internet (IN): padrão.
- Tipo: define o tipo de registro.
- Dados: dados específicos para o tipo de registro de recurso.
  - Exemplo: tipo A contém um endereço; tipo MX contém prioridade e endereço.

Os registros do banco de dados do DNS são formados por cinco campos:

- Nome;
- Time To Live (TTL);
- Classe;
- Tipo;
- Dados.



```
<nome><TTL><classe><tipo><dados>
```

Exemplo:

```
www 604800 IN A 200.130.77.75
```

| Tipo  | Descrição                        | Valor                            |
|-------|----------------------------------|----------------------------------|
| SOA   | Início de autoridade             | Informações do domínio e da zona |
| A     | Mapeamento de nome para endereço | Endereço IPv4 (32 bits)          |
| PTR   | Mapeamento de endereço para nome | Alias para endereço IP           |
| MX    | Servidor de correio eletrônico   | Domínio e prioridade             |
| NS    | Servidor de nomes                | Nome de um servidor de nomes     |
| CNAME | Nome canônico (para alias)       | Nome no domínio                  |
| HINFO | Informações sobre o computador   | Recursos de hardware e software  |
| TXT   | Informações textuais             | Informação de uso geral          |

**Tabela 3.1**  
Tabela de tipos de registros.

Relembrando: registro de recurso é um conjunto de cinco campos, codificados normalmente em ASCII, com cada registro ocupando uma linha do arquivo.

Cada registro é composto por: name, Time To Live, type, class e value. Onde *name* é o nome do recurso (que pode ser um computador, uma impressora etc.), TTL é o tempo de duração da validade da informação em segundos, *type* é o tipo de registro, *class* indica a que classe a informação pertence e *value* é o conteúdo, que depende do parâmetro *type*. O parâmetro *type* pode ser:

- **Start Of Authority (SOA):** contém informações sobre o domínio e zona;
- **Address (A):** endereço IPv4 com 32 bits;
- **Pointer (PTR):** associa um endereço IP a um nome;
- **Mail eXchange (MX):** identifica o servidor de correio eletrônico;
- **Name Server (NS):** nome de um servidor de DNS no domínio;
- **Canonical NAME (CNAME):** nome do recurso no domínio;
- **Hardware INfOrmation (HINFO):** informações pertinentes ao hardware;
- **TeXT (TXT):** informações gerais.

Além desses parâmetros, existem outros menos utilizados:

- **AAAA:** mapeia nomes para endereços IPv6 com 128 bits;
- **ISDN:** mapeia nomes para números de telefone.

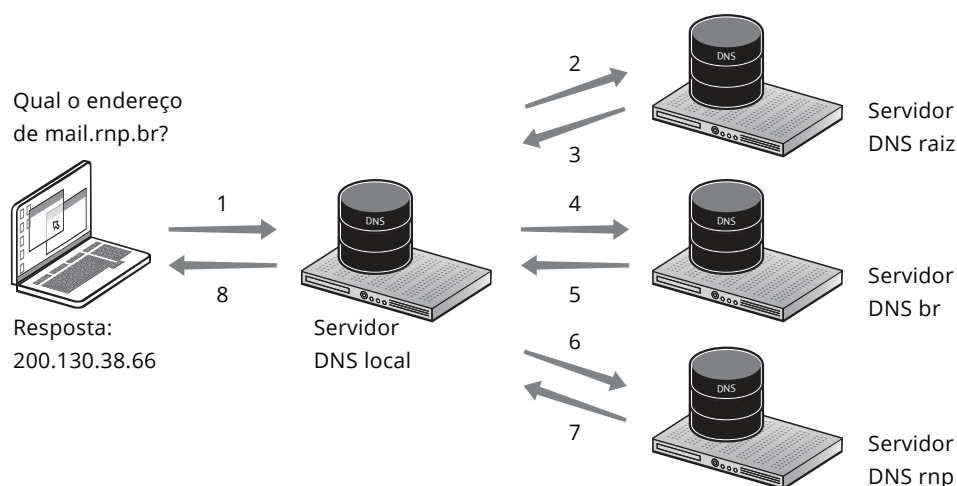
Todo domínio, sendo um simples host ou um domínio de nível mais alto, pode ter um conjunto de registros de recurso associado a ele. Para um host qualquer, o registro de recurso mais comum é somente seu IP, embora existam muitos outros tipos de registros de recurso. Quando um *resolver* envia um nome de domínio ao DNS, o que volta é o registro de recurso associado com aquele nome. Assim, a verdadeira função de DNS é mapear nomes de domínio sobre registros de recurso.

Os tipos de dados definem o tipo de registro, o que muitas vezes permite identificar o tipo de servidor.

- **SOA:** identifica o servidor Start of Authority do domínio;
- **A e PTR:** identificam servidores utilizados respectivamente em mapeamento direto e reverso;
- **MX:** identifica um servidor de correio eletrônico do domínio. Deve ser acompanhado de um número de prioridade caso exista mais de um servidor de correio eletrônico no domínio;
- **NS:** indica o servidor de nomes do domínio;
- **CNAME:** indica nome canônico, o que permite a um determinado servidor responder por mais de um nome;
- **HINFO:** indica informação sobre o hardware e software; pouco utilizado na prática;
- **TXT:** indica informações de uso geral.

## Mapeamento direto e reverso

O mapeamento direto é utilizado na tradução de nomes em endereços IP, que será utilizado como destinatário do pacote a ser transmitido pela rede.



**Figura 3.2**  
Mapeamento direto.

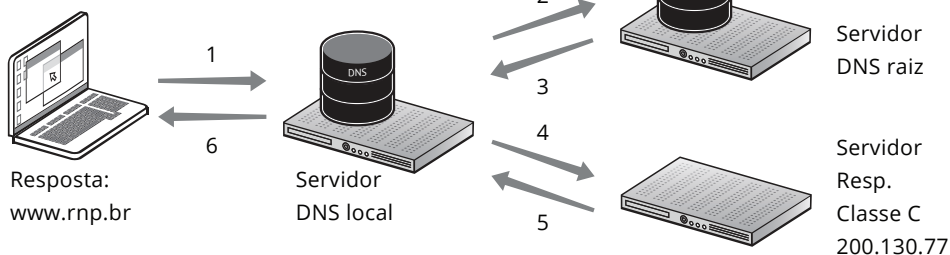
A consulta segue um entre dois processos:

- O computador do usuário encaminha uma consulta ao servidor DNS presente em sua rede local. Como este não tem a informação em seu banco de dados, procederá com várias consultas até a determinação do endereço IP associado ao nome informado. Em última instância, quem informará o endereço IP associado é o servidor DNS da rede de destino. Todavia, para localização do servidor DNS da rede de destino, é feita uma consulta a um servidor raiz da internet para a determinação do domínio de topo. Em seguida é feita uma consulta sobre a máquina responsável pelo DNS do domínio procurado. Por último é feita uma consulta ao servidor DNS da rede de destino, que por sua vez informará o endereço IP da máquina desejada.
- O servidor DNS local já possui em cache o endereço procurado. Nesse caso, a resposta é enviada diretamente ao computador que fez a consulta, sem necessidade de encaminhar consultas a outros servidores.

O mapeamento reverso é utilizado na tradução de endereços IP em nomes, na consulta encaminhada ao servidor DNS da rede local. O servidor DNS local consulta um servidor DNS raiz e em seguida o servidor DNS na rede de destino. O nome é então passado ao computador que efetuou a consulta.



Qual o nome da máquina com IP 200.130.77.75?



**Figura 3.3**  
Mapeamento reverso.

## Cliente e servidor DNS

*Resolver* (resolvedor): cliente.

- Efetua a montagem das consultas.
- Biblioteca de rotinas link-editada a qualquer aplicação que deseja ter um endereço traduzido.
- Usa o arquivo `/etc/resolv.conf` para localizar o servidor DNS.

*Named*: servidor de nomes.

- Daemon (serviço) responde às consultas efetuadas.

O software de serviços de nomes do DNS é dividido conceitualmente em dois componentes:

- **'Resolver' (lado cliente)**: módulo de software responsável pela montagem das consultas;
- **Named (servidor de nomes)**: processo responsável por responder às perguntas, fornecendo as respostas apropriadas.

A implementação mais comumente encontrada do DNS, tanto para o *resolver* quanto para o servidor de nomes, é chamada Berkeley Internet Name Domain Server (BIND) para ambientes Unix.

O *resolver* não existe como um processo distinto executado no computador. Ele é, na realidade, uma biblioteca de rotinas de software (chamadas código do *resolver*) que é ligada (link-editada) a qualquer aplicação que deseja traduzir endereços. Essa biblioteca sabe como solicitar do servidor de nomes informações a respeito de uma determinada estação. Do ponto de vista de uma aplicação, o acesso ao DNS se dá pelo uso de um módulo de software chamado *resolver*, que faz parte da própria aplicação, isto é, ele não faz parte do núcleo do Sistema Operacional (já os protocolos TCP/IP são ligados ao núcleo). Os protocolos TCP/IP no núcleo não conhecem nada a respeito do DNS. Uma aplicação (ou um serviço não transparente TCP/IP) precisa traduzir o endereço simbólico de seu computador hospedeiro para um endereço IP antes de poder iniciar uma conexão de transporte (TCP ou UDP), e é o *resolver* que faz essa tradução. O *resolver* se comunica como os servidores de nomes, geralmente, por meio de conexões UDP. Para efetuar a tradução, o resolver contata um ou mais servidores de nomes na rede, conforme indicado no arquivo `/etc/resolv.conf`.



Esse tipo de consulta é usado frequentemente em diagnósticos e na localização de um spammer ou hacker.

## Servidor DNS

- Quando um resolvedor efetua uma consulta, esta é enviada a um servidor de nomes local.
- Se o domínio consultado estiver sob a jurisdição do servidor de nomes, este retornará os dados oficiais.





- Ou consultará o cache, que armazena as consultas efetuadas recentemente.
  - ▣ As informações no cache podem estar desatualizadas.

Servidores DNS fazem a tradução de nomes de domínio em números IP. Isso pode parecer um trabalho simples, e seria, se não fosse por quatro fatores:

- Existem bilhões de endereços IP atualmente em uso, e a maioria das máquinas possuem um nome;
- A quantidade de solicitações feitas aos servidores DNS alcança a casa de muitos bilhões em apenas um dia;
- Nomes de domínio e números IP sofrem alterações diárias;
- Novos domínios são criados diariamente.

O sistema DNS é um imenso banco de dados, e nenhum outro banco de dados recebe tantas solicitações. Nenhum banco de dados no planeta tem milhões de pessoas modificando seu conteúdo, diariamente. Esses fatores é que fazem o serviço DNS ser tão especial.

O daemon em execução no servidor DNS “escuta” na porta 53 por consultas feitas a outros computadores. O servidor procura localmente em sua base de dados ou encaminha a consulta a outros servidores (raiz, domínio de topo e domínio).

Caso o servidor responsável pelo DNS no domínio seja consultado, retornará os dados oficiais.

## Tipos de servidores DNS

Existem três tipos de servidores DNS.



- Servidor primário (master):
  - ▣ Mantém as informações completas sobre os domínios que administra.
  - ▣ Atualizações devem ser feitas no servidor primário.
- Servidor secundário (slave):
  - ▣ Mantém cópias dos dados disponibilizados no servidor primário.
  - ▣ Atua mesmo com o servidor primário indisponível.
  - ▣ Podem existir múltiplos servidores secundários.
- Servidor cache (catching):
  - ▣ Não mantém informações sobre os domínios, apenas repassa as solicitações recebidas para outros servidores remotos.
  - ▣ Mantém respostas de consultas efetuadas anteriormente.
  - ▣ Os dados possuem tempos de vida (TTL) para possibilitar o descarte de valores desatualizados.
  - ▣ Utilizados também para aumento de disponibilidade e desempenho, aliviando a carga sobre os servidores primário e secundários.

Com o BIND, cada estação usa o código do *resolver*, porém nem todas as estações executam o processo servidor de nomes. Uma estação que não executa um processo servidor de nomes localmente e depende de outras estações para conseguir as respostas para seus serviços de nomes é chamada sistema resolvidor (resolver-only system). As configurações somente resolvidoras são comuns apenas em estações sem disco (diskless). A grande maioria das estações executa localmente um processo servidor de nomes.

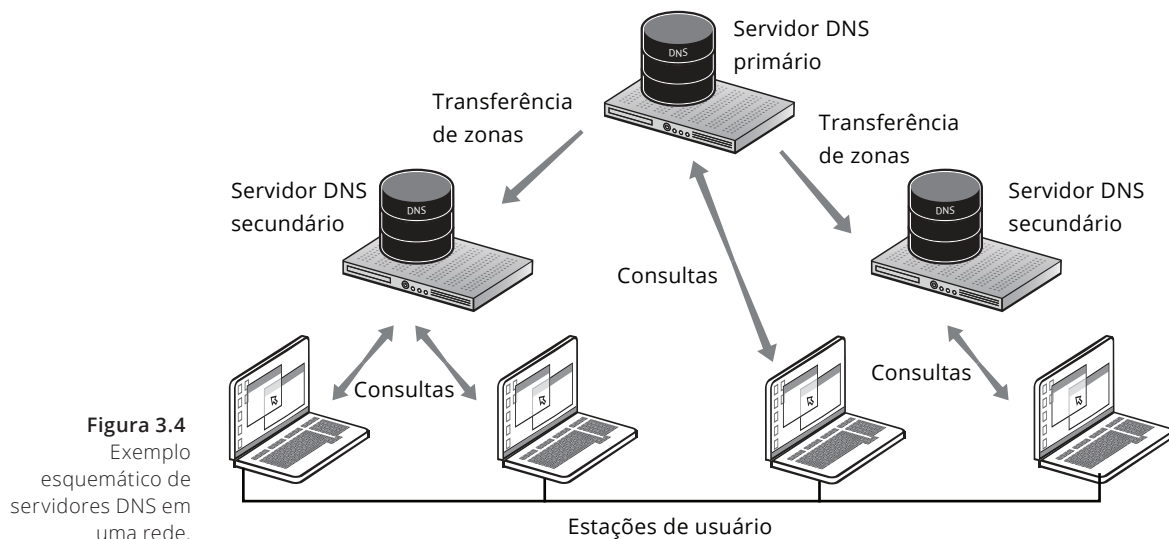


O servidor de nomes do BIND é executado como um processo distinto chamado *named* (daemon). Os servidores de nomes são classificados de diferentes maneiras, dependendo do modo como estejam configurados. As três principais categorias de servidores de nomes são:

- **Servidor Primário (master):** servidor a partir do qual todos os dados a respeito de um domínio são originados. O servidor primário faz a carga das informações a respeito do domínio diretamente a partir de um arquivo de especificações, no formato texto, criado pelo administrador de sistemas. Os servidores primários são autorizados, isto é, têm informações completas e atualizadas a respeito de seus domínios. Só pode existir um único servidor primário para cada domínio. Em outras palavras, esse servidor executa o serviço DNS e é a fonte autorizada de informações sobre um domínio, obtendo informações sobre o domínio de arquivos locais que devem ser mantidos pelo seu administrador;
- **Servidor Secundário (slave):** servidores para os quais uma base de dados completa, referente a um determinado domínio, é obtida a partir do servidor primário. Uma base de dados correspondente a um determinado domínio, replicada no servidor secundário, é chamada de arquivo de zona. Copiar uma base de dados de um servidor primário para um servidor secundário garante que ele sempre terá informações atuais sobre um domínio por meio de transferências periódicas de arquivos de zona para esse domínio. Os servidores secundários são autorizados somente para seus domínios. Em outras palavras, esse servidor executa o servidor DNS, mas obtém informações sobre o domínio que está servindo a partir do servidor primário. Atende a consultas DNS mesmo que o servidor primário esteja indisponível, atuando neste caso como um servidor de backup.
- **Servidor cache (catching):** servidores recebem as respostas para todas as consultas de serviços de nomes que venham de outros servidores de nomes. Quando um servidor cache recebe uma resposta para uma consulta, ele guarda essa informação para utilizá-la em outras consultas que venham a ocorrer. A maioria dos servidores de nomes armazena respostas em memória cache e as utiliza dessa maneira. O que torna os servidores de *catching* importantes é que essa técnica é a única utilizada na construção de sua base de dados para o domínio correspondente.

Servidores cache não são autorizados, isto é, suas informações podem estar incompletas ou desatualizadas, apesar de geralmente estarem coerentes. Atuam somente para armazenamento de consultas prévias. Sua principal função em uma determinada rede ou sub-rede é aumentar a confiabilidade e o desempenho de consultas DNS, aliviando a carga sobre os servidores primário e secundários da rede.

Uma boa prática, em redes de grande porte, é a utilização de um servidor primário apoiado por um ou dois servidores secundários para toda a rede, colocando servidores de cache em determinadas sub-redes para aumento de desempenho e confiabilidade, conforme mencionado.



## Servidores raiz DNS

- A zona de root é o nível mais alto na hierarquia do sistema DNS.
- Essa zona acomoda a identificação dos 13 servidores raiz de DNS.
- Mantida pela Internet Assigned Numbers Authority (IANA), vinculada à Internet Corporation for Assigned Names and Numbers (ICANN).
- Todo servidor DNS deve possuir uma referência aos servidores raiz, utilizados como ponto inicial de consulta pelos servidores DNS da rede.

O serviço DNS é essencial ao funcionamento dos principais protocolos utilizados na internet. Como se trata de um ponto crítico para toda a rede mundial de computadores, esses treze servidores estão espalhados pelo planeta.

Cada um desses treze servidores é de responsabilidade de um operador, que os arranja em sua maioria em conjunto de múltiplos servidores, denominados clusters, localizados muitas vezes em diferentes cidades e países.

## Estrutura do DNS no Linux

Servidor:

- Executável.
- Arquivo de configuração *named.conf*.
- Arquivo de mapas da rede (zonas).
- Arquivo para localização do servidor DNS (*resolv.conf*).

Cliente:

- Arquivo para localização do servidor DNS (*resolv.conf*).

Nos Sistemas Operacionais Linux, a estrutura existente funciona com servidores e clientes. Nos servidores ficam todos os arquivos que compõem o banco de dados DNS: configuração (*named.conf*), mapas de rede e o *resolv.conf*, que aponta para os servidores DNS.

Nos clientes, apenas o arquivo *resolv.conf* é necessário. Um servidor de DNS tem como objetivo fazer a conversão entre nomes de computadores e seus números IP. O programa (daemon) responsável pela tarefa é o */usr/bin/named*, que é parte do pacote BIND.

Leia a relação  
atualizada no site  
<http://www.root-servers.org>

Para o funcionamento correto do DNS, alguns arquivos são necessários para a sua configuração. Os arquivos podem ter qualquer nome, mas devem estar definidos em *named.conf* (aqui estamos usando os padrões da internet).

- **/etc/bind/named.conf**: principal arquivo de configuração do servidor DNS;
- **/etc/bind/db.root**: arquivo que aponta para hosts externos, essencial quando o servidor estiver sendo utilizado na internet;
- **etc/bind/db.local**: responsável pela resolução direta do domínio localhost;
- **/etc/bind/db.127**: responsável pela resolução reversa do domínio 127.0.0.0;
- **/etc/bind/db.empresa.com.br**: responsável pela resolução direta do domínio empresa.com.br;
- **/etc/bind/db.X.168.192**: responsável pela resolução reversa do domínio empresa.com.br, supondo que a rede possui endereço de rede 192.168.X.0/255.255.255.0.

A estrutura do serviço DNS no Linux, assim como em outros sistemas, consiste em uma arquitetura cliente/servidor. No servidor há alguns arquivos, como:

- **named**: arquivo executável (daemon);
- **named.conf**: arquivo de configuração lido pelo executável no momento de sua inicialização;
- **Arquivos de mapas da rede**: arquivos com as informações sobre o(s) domínio(s) controlado(s) pelo servidor;
- **resolv.conf**: arquivo utilizado pelo resolvedor, presente tanto no servidor quanto nas estações Linux.

## Arquivo de configuração 'named.conf'

O arquivo *named.conf* possui a seguinte estrutura:

- Opções default.
- Definição das zonas.
  - ▣ Cache.
  - ▣ Loopback reverso.
  - ▣ Domínio direto.
  - ▣ Domínio reverso.



Esse é o arquivo que contém as informações referentes a cada uma das zonas administradas.

### Arquivo de configuração DNS: *named.conf*

```
*****

options {
    directory "/var/named";
    forwarders {
        200.221.11.100;
        200.221.11.101;
    };
};

zone "." IN {
    type hint;
    file "caching-example/named.ca";
};
*****
```

```

);

zone "localhost" IN {
    type master;
    file "caching-example/localhost.zone";
    allow-update { none; };
};

zone "0.0.127.in-addr.arpa" IN {
    type master;
    file "caching-example/named.local";
    allow-update { none; };
};

zone "site.local" IN {
    type master;
    file "site.local.zone";
    allow-transfer { 192.168.0.1; };
};

zone "rnp.br" IN {
    type slave;

    file slave.rnp.br;

    masters { 200.130.77.69; };
};

*****

```

- **options:** descreve as opções de inicialização do BIND;
- **directory:** diretório onde ficam os arquivos de banco de dados do domínio;
- **forwarders:** endereços IP dos servidores utilizados para solicitações DNS quando o servidor não souber resolver;
- **zone:** zona que será configurada;
- **type:** o tipo de aplicação da zona no seu servidor DNS;
- **master:** servidor primário;
- **slave:** servidor alternativo;
- **file:** caminho do arquivo de banco de dados do domínio situado na pasta `/var/named`;
- **allow-update:** especifica quais dos servidores da zona serão atualizados. No caso, nenhum;
- **allow-transfer:** servidores que possuem a permissão de serem "slaves". Assim, evita plágio de DNS.

O arquivo `/etc/bind/named.conf` é o principal arquivo do BIND e é o responsável pelas informações usadas para que o serviço seja realizado. O arquivo possui informações referentes ao diretório onde estão os arquivos de dados do DNS, os servidores secundários, as zonas atendidas, entre outras informações pertinentes.

O arquivo de configuração `named.conf` possui uma série de cláusulas que são lidas pelo daemon. Essas cláusulas são estruturas que agrupam *statements* relacionadas, que são como diretivas

que norteiam o controle do comportamento do servidor em relação a várias situações.

Relação de cláusulas que podem ser utilizadas:

- **Access Control List (ACL):** define uma ou mais listas de controle de acesso, grupo de hosts e usuários;
- **Controls:** descreve e controla o acesso ao canal de controle utilizado pelo administrador com a chave `rndc`;
- **include:** não é propriamente uma cláusula, mas referência a outros arquivos onde podem ser colocadas cláusulas que influenciam no funcionamento do servidor, e que estão em separado normalmente por questões de segurança ou facilidade de administração;
- **key:** define chaves utilizadas em operações que exigem autenticação;
- **logging:** define o nível de log e a localização dos arquivos de log;
- **lwres:** propriedades do BIND quando usa um resolvidor leve;
- **options:** *statements* para controle de opções genéricas ou globais, que podem ser sobrepostas por configurações nos arquivos de zona;
- **server:** define o comportamento do servidor ao responder ou acessar um servidor remoto;
- **trusted-keys:** define opções para acessos que exigem autenticação;
- **view:** controla as funcionalidades do BIND em função do endereço do host;
- **zone:** define as zonas mantidas pelo servidor.

As informações disponibilizadas por meio dos *statements* no arquivo de configuração podem ainda ser consideradas em razão da função que controlam:

- **queries:** controlam o comportamento das queries;
- **transfer:** controlam o comportamento das transferências de zonas;
- **operations:** controlam o comportamento do servidor;
- **security:** controlam o comportamento do servidor em relação à segurança;
- **statistics:** controlam o comportamento da parte de log do servidor.

Uma lista completa dos *statements* pode ser encontrada na documentação do BIND 9.

## Arquivos de mapas da rede

- Exemplo de arquivo de mapa de rede:
- Zona de domínio direto: rede exemplo.com

```
; arquivo db.zone
$TTL 604800
@      IN      SOA      mar.exemplo.com. root.mar.exemplo.com. (
                                2              ; Serial
                                604800 ; Refresh
                                86400  ; Retry
                                2419200; Expire
                                604800 )      ; Negative Cache TTL
@      IN      NS       mar.exemplo.com.
localhost      IN      A       127.0.0.1
mar            IN      A       200.130.77.130
sol            IN      A       200.130.77.131
```



Esse é o arquivo que contém os nomes dos números IP de cada máquina desse domínio.

### Arquivo de configuração DNS: mapas de rede

```
*****
;
; Configurações do domínio site.local.
;
$TTL 86400
@ IN SOA ns01.site.local. root. (
                                2007071500; Serial
                                28800 ; Refresh
                                14400 ; Retry
                                3600000 ; Expire
                                86400 ; Minimum

604800) ; TTL
;
; Definição dos Servidores de Nome.
;
    IN NS ns01.site.local.
    IN NS ns02.site.local.
;
; Definição dos Mail Exchanger.
;
    IN MX 10 mail.site.local.
;
; Definição dos endereços de Hosts.
;
localhost IN A 127.0.0.1
ns01      IN A 192.168.0.254
ns02      IN A 192.168.0.1
mail      IN A 192.168.0.10

mysql     IN A 192.168.0.9
www       IN A 201.26.142.72
*****
```

- **Serial:** número serial da zona, deve ser incrementado sempre que for feita alguma alteração, para que os servidores secundários possam se atualizar;
- **Refresh:** número de segundos entre pedidos de atualização oriundos dos servidores secundários;
- **Retry:** número de segundos que os servidores secundários vão esperar para refazer uma consulta que falhou;
- **Expire:** número de segundos que um servidor, *master* ou *slave*, esperará para considerar a informação expirada, se ele não conseguir alcançar o servidor de nomes primário;
- **Minimum:** é o TTL default caso o domínio não especifique um TTL;





- **Time to Live (TTL):** número de segundos que um nome de domínio é cacheado localmente antes de expirar e retornar para os servidores de nomes de autoridade para atualização das informações.

Conforme mencionado anteriormente, o funcionamento do servidor DNS é baseado em zonas controladas por mapas de rede. Os arquivos de mapas de rede são os arquivos que contêm as informações de cada uma das zonas atendidas pelo servidor DNS. São de dois tipos: tradução direta e tradução reversa. São os arquivos onde cada um dos computadores que pertencem a uma empresa são descritos, com nome, endereço IP e servidor de e-mail, entre outras informações, como Time To Live (TTL), serial para controle de alterações etc.

Um servidor DNS necessita de pelo menos quatro arquivos com mapas de rede:

- **root.servers:** mantém uma relação dos servidores raiz da internet; normalmente ficam em um arquivo denominado *named.ca*;
- **localhost:** permite a resolução do nome localhost para o endereço de loop local 127.0.0.1;
- **direct-map:** apoiam o processo de conversão de um nome em um endereço IP;
- **reverse-map:** apoiam o processo de conversão de um endereço IP em um nome.

## Exemplo de arquivo de mapa de rede

Zona de domínio reverso: rede 200.130.77:

```
; arquivo db.77.130.200
$TTL 604800
@ IN SOA mar.exemplo.com. root.mar.exemplo.com. (
    2 ; Serial
    604800 ; Refresh
    86400 ; Retry
    2419200 ; Expire
    604800 ) ; Negative Cache TTL
;
@ IN NS mar.exemplo.com.
130 IN PTR mar.exemplo.com.
131 IN PTR sol.exemplo.com.
```

Arquivos com os mapas de rede:

- root.servers (arquivo db.root): servidores raiz da internet;
- localhost (arquivo db.local): informações sobre o nome localhost;
- direct-map (arquivo db.zone): informações sobre o domínio da rede;
- reverse-map (arquivo db.77.130.200): informações sobre a rede 200.130.77.0.

## Arquivo 'resolv.conf'

O arquivo */etc/resolv.conf* é usado pelas máquinas Linux para localizar o servidor DNS.

- Diretivas:
  - nameserver.
  - Endereço IP do servidor DNS.

- ▣ **search.**
  - ▣ Especifica os domínios a serem utilizados na busca pelo host a que se deseja conectar.
  - ▣ A adição de muitos domínios pode causar lentidão na resolução dos nomes.
- ▣ **domain.**
  - ▣ Alternativa à diretiva *search*.

Esse é o arquivo responsável por indicar quem são os servidores de nome para consultas de DNS. Ele contém uma lista de domínios ao qual pertence a máquina e os servidores de nome que serão consultados.

#### Arquivo de configuração DNS – *resolv.conf*

```
*****
search site.local
nameserver 192.168.0.254
nameserver 192.168.0.1
*****
```

- ▣ **search:** especifica os domínios aos quais a máquina pertence;
- ▣ **nameserver:** especifica o endereço IP dos servidores DNS;
- ▣ **domain:** alternativa ao *search*.

Adicionalmente a essas zonas, outras podem ser incluídas. Devem ser incluídas zonas para cada rede ou domínio adicional sob responsabilidade do servidor, além de zona para lidar com servidor secundário. O arquivo */etc/resolv.conf* é o arquivo que possui a informação do computador que está rodando o serviço DNS e o domínio a ser utilizado.

O arquivo *resolv.conf* é utilizado pelo resolvedor (resolver) das máquinas Linux para localizar o servidor DNS. A principal informação desse arquivo é o endereço IP do servidor DNS, informado pela diretiva *nameserver* <Endereço IP>.

Pode ainda ser utilizada a diretiva *search* ou *domain*, útil em consultas DNS quando é informado somente o nome do host em vez do endereço completo (FQDN). Vários domínios podem ser adicionados, o que deve ser evitado, pois causará uma consulta DNS em cada um deles.

### Exercício de fixação 2

#### Servidor de DNS Cache

Instale e configure o servidor de DNS apenas com a função de cache.

#### DNS com IPv6

Com o desenvolvimento do IPv6, surgiu a necessidade de se prover um serviço de nomes que suporte esse novo protocolo. Foi criado um novo registro que armazena endereços no formato do IPv6.

- ▣ **AAAA.**

Um novo domínio foi criado para a resolução de reverso, descrito na RFC 3152.

- ip6.arpa.

Dois tipos de consultas podem ser feitas em um DNS:

- Resolução de nomes.
- Resolução de endereço reverso.

Com o desenvolvimento do IPv6, surge a necessidade de se prover um serviço de nomes que suporte esse novo protocolo. Para isso, foi criado um novo registro que armazena endereços no formato do IPv6, o AAAA. Estão em fase de desenvolvimento novos tipos de registros que facilitarão a manutenção desse serviço. Além disso, um novo domínio foi criado para a resolução de reverso, o ip6.arpa, descrito na RFC 3152.

Como servidor de nomes, pode-se utilizar o BIND versão 9, cuja implementação é a mais robusta. Dois tipos de consultas podem ser feitas em um DNS:

- **Resolução de nomes:** name to address lookups (forward lookups);
- **Resolução de endereço reverso:** address to name lookups (reverse lookups).

## Resolução de Nomes

Para Resolução de Nomes (forward lookups), é usado o tipo AAAA. Os registros AAAA são paralelos e semelhantes aos Registros A no IPv4. Em um único endereço é possível especificar um endereço IPv6 completo. Por exemplo:

```
www.ipv6.br.      IN A      200.160.4.22
www.ipv6.br.      IN AAAA    2001:12ff:0:4::22
```

## Resolução de Endereço Reverso

Para resolução de reverso, foi adicionado o registro PTR ip6.arpa, responsável por traduzir endereços IPv6 em nomes. Em sua representação, omitir sequência de zeros não é permitido e o bit menos significativo é colocado mais à esquerda.

O arquivo de configuração *named.conf* deve ser editado e incluída uma nova zona, a reversa, procedendo da seguinte forma com a configuração:

### Arquivo: *named.conf*

```
*****

zone "0.0.0.0.1.2.f.f.1.0.0.2.ip6.arpa" {
    type master;
    file "2001:12ff:0000.ip6.arpa";
};

*****
```

Passamos, assim, a responder pelo reverso do bloco que foi cedido. Tal prefixo já deve ter sido delegado pelo provedor de endereços IPv6, da mesma forma como ocorre na delegação de subdomínios.

A forma como se declara o registro reverso é semelhante ao IPv4, ou seja, começando pelo último dígito hexadecimal do endereço, e é denominado de "nibble format".



### Arquivo: 2001:12ff:0000.ip6.arpa

```
*****
$TTL 86400
@ IN SOA ns.ipv6.br. root.ipv6..br. (
    2001112201
    3H ; refresh
    15M ; retry
    1W ; expiry
    1D ) ; minimum
    IN NS ns.ipv6.br.
2.2.0.0.0.0.0.0.0.0.0.0.0.4.0.0.0 IN PTR www.ipv6.br.
*****
```

## DNSSEC

- Domain Name System SEcURITY Extensions (DNSSEC) tem por objetivo adicionar mais segurança ao sistema de resolução de nomes.
- Passou a ser um padrão internacional que estende a tecnologia DNS, garantindo a autenticidade e a integridade das informações.
- Utiliza o mecanismo de criptografia de chaves públicas.
- Criado para solucionar alguns problemas encontrados no atual sistema de resolução de nomes.
- O objetivo da extensão DNSSEC é assegurar o conteúdo do DNS e impedir os ataques.

Tem por objetivo adicionar mais segurança ao sistema de resolução de nomes, reduzindo o risco de manipulação das informações por terceiros. Passou a ser um padrão internacional, que estende a tecnologia DNS, garantindo a autenticidade e a integridade das informações. Utiliza o mecanismo de criptografia de chaves públicas. Foi criado para solucionar alguns problemas encontrados no atual sistema de resolução de nomes, como as falsas resoluções que criam oportunidades para alterações de dados em diversos tipos de transações, como por exemplo, compras eletrônicas. No protocolo DNS, um ataque onde a informação é corrompida é extremamente difícil de ser detectado e, na prática, impossível de ser prevenido. O objetivo da extensão DNSSEC é assegurar o conteúdo do DNS e impedir estes ataques, validando os dados e garantindo a origem das informações.

- Provê segurança para a resolução de endereços.
- Funciona como um caminho alternativo para a verificação de autenticidade da informação.
- Sua verificação ocorre antes de outras aplicações de segurança, tais como, ssl, ssh, sftp, PGP etc.
- Permite que se obtenha com precisão o endereço IP, por meio dos seguintes serviços:
  - ▣ Autenticação da origem.
  - ▣ Integridade dos dados.
  - ▣ Prova de inexistência.



O sistema DNSSEC permite que se obtenha com precisão o endereço IP, através dos seguintes serviços:

- **Autenticação da origem:** a informação somente pode ser originada do servidor DNS autoritativo do domínio;
- **Integridade dos dados:** a informação recebida é exatamente a mesma fornecida pelo servidor DNS autoritativo do domínio;
- **Prova de inexistência:** no caso de uma resposta negativa fornecida pelo servidor DNS autoritativo do domínio, pode-se comprovar que essa informação está correta. Dependendo do tipo de ataque, isso pode ser suficiente para que o sistema DNSSEC o interrompa.

- Para que esses serviços possam ser ativados, o DNSSEC necessita que todas as solicitações sejam assinadas digitalmente.
- Novos Registros de Recursos passam a ser criados durante o processo de assinatura digital:
  - DNSKEY.
  - RRSIG.
  - Delegation Signer (DS).
  - NSEC.



Para que esses serviços possam ser ativados, o DNSSEC necessita que todas as solicitações sejam assinadas digitalmente, usando uma ou mais chaves privadas e um algoritmo assimétrico de criptografia. Novos Registros de Recursos (RRs) passam a ser criados durante o processo de assinatura digital, usando o utilitário `dnssec-signzone`.

- **DNSKEY:** armazena as chaves públicas utilizadas para a assinatura do domínio;
- **RRSIG:** armazena as chaves privadas utilizadas para a assinatura dos Rrset;
- **Delegation Signer (DS):** ponteiro para a próxima cadeia de confiança;
- **Next Secure (NSEC):** permite autenticar uma resposta negativa.

## DNSKEY

É um Resource Record que armazena a chave pública da zona de autoridade.

```
exemplo.com.      900          IN      DNSKEY 256 3 5 (
    AwEAAeZPN2yMs9q6kgYjFUb1EwjCnWWcPq+TGcJrD5ga
    XXAbP5MAqIkGZ5J4TU1mmpL1A8gMfd/wUmBkVipXR8FK
    HRajBZSRfgeKnKaQtrxNZ32Ccts2F6Y1v9WaLXtiqebg
    0ZtuJFpQr6pnIt/Fo0I+I7BUSNrX28VTq4jXu/qTrmM/ ); key
id = 62745
```

## RRSIG

É um Resource Record que contém a assinatura de um RRset (conjunto de RR com o mesmo nome de domínio, classe e tipo) específico de uma determinada chave DNSKEY. Possui uma validade inicial e final.

```

exemplo.com.      900 IN RRSIG SOA 5 3 900 20070617200428 (
                  20070518200428 62745 mar.exemplo.com
                  glEeCYyd/CCBfzH64y0RAQf90xYDsI4xuBNaam+8DZQZ
                  xeoSLQEEtwmp6wBtQ7G10wSM9nEjRRhbZdNPNKJMp2PE
                  lLLgLI+BLwd1z0t8MypcpL0aTm9rc7pP7UR5XLzU1k8D
                  m6ePW1bNkId7i0IPSghyoHM7tPVdL2GW51hCuJA= )

```

## Delegation Signer (DS)

É um hash do registro de recurso DNSKEY. Serve para informar a existência de uma cadeia de confiança entre um domínio e seus subdomínios. Indica que a zona delegada está assinada e qual a chave que foi utilizada.

O Registro de Recurso DS é um ponteiro para a cadeia de confiança, que garante a autenticidade das delegações de uma zona até um ponto de confiança.

```

exemplo.com. IN DS 817 5 1 EAEC29E4B0958D4D3DFD90CC70C6730AD5880DD3

```

É possível obter os DS da zona utilizando o sistema Whois.

```

$whois rnp.br
% Copyright (c) Nic.br
% The use of the data below is only permitted as described in
% full by the terms of use (http://registro.br/termo/en.html),
% being prohibited its distribution, comercialization or
% reproduction, in particular, to use it for advertising or
% any similar purpose.
% 2012-02-16 17:42:15 (BRST -02:00)
domain:      rnp.br
owner:       Associação Rede Nacional de Ensino e Pesquisa
ownerid:     003.508.097/0001-36
responsible: Nelson Simões Silva
country:     BR
owner-c:     RC0217
admin-c:     NES
tech-c:      FRK16
billing-c:   RC0217
nserver:     teyr.na-cp.rnp.br 200.130.25.17 2001:12f0:3f:3::11
nsstat:      20120216 AA
nslastaa:    20120216
nserver:     bellana.nc-rj.rnp.br 200.143.193.3 2001:12f0:3e:102::3

```

```

nsstat: 20120216 AA
nslastaa: 20120216
nserver: lua.na-df.rnp.br 200.130.77.69 2001:12f0:3d:102::45
nsstat: 20120216 AA
nslastaa: 20120216
dsrecord: 51124 RSA/SHA-1 4ED9FC74C63C99E52E2FC492517C73AEFAC316F2
dsstatus: 20120216 DSOK
dslastok: 20120216
created: before 19950101
changed: 20111125
status: published

```

## Next Secure (NSEC)

Permite autenticar uma resposta negativa, com pré-assinatura, sem a necessidade de chaves on-line para assinatura, diminuindo a possibilidade de ataque DOS.

```

exemplo.com 900 IN NSEC ns1.exemplo.com. NS SOA RRSIG NSEC DNSKEY

```

## Implementação do DNSSEC

- Utilizar o aplicativo `dnssec-keygen` para a geração das chaves do domínio.
- Usar o aplicativo `dnssec-signzone` para assinar o arquivo de zona do domínio.
- Atualizar o arquivo `named.conf` para referenciar o novo arquivo de zona.
- Reiniciar o Bind.
- Adicionar o DS no site do registro.br.
- Aguardar a nova publicação no site do registro.br.



1. Utilizar o aplicativo `dnssec-keygen` para a geração das chaves do domínio.

```

$ dnssec-keygen -r /dev/urandom -f KSK -aRSASHA1 -b 1024 -ZONE
exemplo.com

```

O comando vai gerar dois arquivos, com as extensões `.key` e `.private`.

2. Usar o aplicativo `dnssec-signzone` para assinar o arquivo de zona do domínio.

```

$ dnssec-signzone -S -z -o exemplo.com db.exemplo.com

```

O comando vai gerar um novo arquivo de zona com a extensão `.signed`. O período de validade das chaves é de 30 dias e as chaves deverão ser revalidadas antes de sua expiração. Para isso, altere o serial no registro SOA e repita o processo.

3. Atualizar o arquivo `named.conf` para referenciar o novo arquivo de zona.

```

zone "exemplo.com"{
    type master;

```

```
file "db.exemplo.com.br.signed;  
);
```

4. Reiniciar o Bind.

5. Adicionar o DS no site do registro.br.

```
$ cat dsset-exemplo.com | head -1  
exemplo.com. IN DS 817 5 1 EAEC29E4B0958D4D3DFD90CC70C6730AD5880DD3
```

Copiar os dados de Key Tag e Digest do arquivo dsset-exemplo.com para a interface no site do registro.br

6. Aguardar a nova publicação no site do registro.br.

### Exercício de fixação 3

#### DNSSEC

São vantagens de se implementar o DNSSEC:

- ☐ Fornece autenticação da origem da informação.
- ☐ Evita ataque do tipo DDOS.
- ☐ Evita ataques do tipo Man-in-the-Middle e Spoofing.
- ☐ Garante a integridade dos dados DNS.
- ☐ Reduz o consumo de energia, sendo boa prática na TI verde.
- ☐ Evitar a manipulação da memória cache (Pharming, Phishing etc.).
- ☐ Garante a integridade dos dados do usuário.
- ☐ É independente dos algoritmos criptográficos.

#### Domínios virtuais

- Um domínio virtual é uma entrada DNS em um servidor que responde por múltiplos endereços IP.
- Necessário caso se deseje que um mesmo servidor DNS responda por vários endereços IP.
- Exemplo:
  - ▣ Permitir que uma única máquina atenda por:
    - ▣ www.virtual1.exemplo.com: 192.168.1.10
    - ▣ www.virtual2.exemplo.com: 192.168.1.20
  - ▣ Utiliza recurso de IP aliasing presente no kernel do Linux.
    - ▣ Algumas vezes é necessário recompilar o kernel e depois configurar normalmente arquivos de mapas da rede.



Domínios virtuais são entradas no DNS para permitir que um servidor possa responder por mais de um número IP; são necessários para que um único servidor DNS responda a solicitações por mais de um número IP.





## Testando o servidor DNS

Para testar o servidor DNS, podem ser utilizados os seguintes comandos:

- *nslookup* (em desuso).
- *dig* (atual).

Comando *dig*:

- Ferramenta flexível para consultas DNS, que apresenta informações detalhadas sobre o endereço consultado.
- Exemplo:
  - `dig -x 200.130.77.75`
  - `dig www.rnp.br`

Exemplo de utilização do comando *dig*:

```
dig www.rnp.br

; <<>> DiG 9.2.4 <<>> www.rnp.br
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53116
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0
;; QUESTION SECTION:
;www.rnp.br.                IN      A
;; ANSWER SECTION:
www.rnp.br.                 290     IN      A      200.143.193.5
www.rnp.br.                 290     IN      A      200.130.77.75
;; Query time: 19 msec
;; SERVER: 172.19.0.2#53(172.19.0.2)
;; WHEN: Fri Sep 29 16:49:21 2006
;; MSG SIZE rcvd: 60
```

## Network File System (NFS)

O NFS foi desenvolvido para permitir que máquinas possam montar uma partição de disco em uma máquina remota como se fosse em um disco local, permitindo o compartilhamento de arquivos na rede. Este capítulo aborda a teoria desse serviço, configuração, otimização de desempenho e ajustes, sendo que ao final serão praticadas a configuração do servidor e do cliente, e ainda testes de funcionamento.



Desenvolvido para permitir que máquinas montem partições de disco em máquinas remotas.

- Compartilhamento na rede.

Outros sistemas:

- Samba.
- AFS.
- Entre outros.

Há outros sistemas que proveem uma funcionalidade similar ao NFS; o Samba, por exemplo, provê serviços de arquivos aos clientes Windows; o Andrew File System (AFS), um projeto open-source da IBM que provê um mecanismo de compartilhamento de arquivos com características de performance e segurança adicional. Algumas das características do AFS foram incluídas na versão 4 do NFS.

Vantagens do NFS: maturidade e sistema padrão suportado por diversas plataformas.

## Configurando um servidor NFS

Instalar:

- `nfs-kernel-server`.
- `nfs-commom`.
- `portmap`.

### Exercício de fixação 4

#### Instale o servidor NFS

Instale o serviço NFS no servidor 192.168.X.1. Para isso, é necessário que estejam instalados três pacotes: `nfs-kernel-server`, `nfs-common` e `portmap`.

Arquivos de configuração:

- `/etc/exports`.
- `/etc/hosts.allow`.
- `/etc/hosts.deny`.

Para configurar um servidor NFS, é necessário fazer a configuração do serviço NFS, por meio de seus arquivos de configuração, e disparar a execução do serviço.

O arquivo `/etc/exports` contém uma lista de entradas; cada entrada indica um volume compartilhado e como cada um está compartilhado. Uma entrada desse arquivo é similar a:

```
directory machine1(option11,option12) machine2(option21,option22)
```

Onde:

- **directory**: indica o diretório que se deseja compartilhar e pode ser um volume inteiro. Ao compartilhar um diretório, todos os diretórios abaixo dele no mesmo sistema de arquivos também serão compartilhados;
- **machine1** e **machine2**: máquinas clientes que terão acesso ao diretório especificado em `directory`. As máquinas podem ser listadas usando endereços IP ou nomes (por exemplo, `machine.company.com` ou `192.168.0.8`). Usar endereços IP é mais confiável e mais seguro;
- **optionxx**: lista de opções para cada máquina, que descreverá o tipo de acesso que a máquina terá. Algumas opções importantes serão examinadas a seguir.

## /etc/exports

Lista entradas, com cada uma indicando um volume compartilhado.

- `directory machine1 (option11, option12).`
- `machine2 (option21, option22).`
- Opções:
  - `ro`
  - `rw`
  - `no_root_squash`
  - `no_subtree_check`
  - `sync`

Opções do arquivo */etc/exports*:

- **ro**: diretório compartilhado apenas para leitura; a máquina cliente não será capaz de escrever. Essa é a opção default.
- **rw**: a máquina cliente terá acesso de leitura e escrita ao diretório;
- **no\_root\_squash**: por default, qualquer pedido feito pelo superusuário na máquina cliente é tratado como se fosse feito pelo usuário *nobody* no servidor. Se *no\_root\_squash* for selecionado, então o superusuário na máquina cliente terá o mesmo nível de acesso aos arquivos no sistema como superusuário no servidor, o que pode ter sérias implicações de segurança, mas pode ser necessário se quisermos executar trabalhos administrativos na máquina cliente que envolvam diretórios exportados;
- **no\_subtree\_check**: se apenas uma parte do volume é exportada, uma rotina chamada *subtree checking* verifica se um dado arquivo requisitado do cliente está na parte adequada do volume. Se o volume inteiro for exportado, desabilitar essa opção aumentará a velocidade de transferências;
- **sync**: por default, os comandos *exportfs* usam o comportamento *async*, indicando para a máquina cliente que um processo de escrita num dado arquivo está completo quando o NFS finalizar a escrita no sistema de arquivos. Esse comportamento pode causar corrupção nos dados, em caso de reinicialização do servidor, o que pode ser prevenido pela opção “sync”.

Um exemplo:

- `/usr/local 192.168.0.1 (ro) 192.168.0.2 (ro)`
- `/home 192.168.0.1 (rw) 192.168.0.2 (rw)`

Outro exemplo:

- `/usr/local 192.168.0.0/255.255.255.0 (ro)`
- `/home 192.168.0.0/255.255.255.0 (rw)`

Para usar netgroups NIS, basta anexar o nome do netgroup com um caractere “@”.

Suponha que temos duas máquinas clientes, *slave1* e *slave2*, que têm endereços IP 192.168.0.1 e 192.168.0.2, respectivamente, e que queremos compartilhar binários e alguns diretórios com essas máquinas. Uma configuração típica para */etc/exports* pode ser:

```
/usr/local    192.168.0.1 (ro) 192.168.0.2 (ro)
/home         192.168.0.1 (rw) 192.168.0.2 (rw)
```

Sendo assim, há um compartilhamento read-only para slave1 e slave2, porque eles devem provavelmente conter os binários. Por outro lado, alguns diretórios necessitam ser exportados read-write, se os usuários quiserem efetuar escritas sobre eles.

Ao se tratar com uma rede de grande porte, primeiro pode-se acessar uma grande faixa de máquinas especificando uma rede e uma máscara. Por exemplo, se quiser permitir o acesso para todas as máquinas com endereços IP entre 192.168.0.0 e 192.168.0.255, então podemos ter as seguintes entradas:

|            |                                |
|------------|--------------------------------|
| /usr/local | 192.168.0.0/255.255.255.0 (ro) |
| /home      | 192.168.0.0/255.255.255.0 (rw) |

Para especificar um netgroup NIS no seu arquivo *exports*, simplesmente anexe o nome do netgroup com um caractere "@".

Outro formato que pode ser utilizado, em vez de hostnames, é algo do tipo \*.foo.com ou 192.168.

Algumas precauções devem ser tomadas em relação ao que pode e o que não pode ser exportado. Primeiro, se um diretório é exportado, seus diretórios pai e filho não podem ser exportados se eles estiverem no mesmo sistema de arquivos. Porém, a exportação de ambos os diretórios pode não ser necessária, uma vez que listar o diretório pai no arquivo */etc/exports* fará com que todos os diretórios sejam exportados, dentro do sistema de arquivo.

## */etc/hosts.allow* e */etc/hosts.deny*

- Especificam os hosts que podem usar serviços numa dada máquina.
- *hosts.allow* é analisado antes do servidor responder a um pedido
- Se a máquina não se associa a uma entrada em *hosts.allow*, o servidor então verifica *hosts.deny*.



Esses dois arquivos são utilizados para controlar o acesso aos serviços TCP do seu computador, especificando as máquinas que, na rede, podem ou não usar esses serviços. Cada linha dos arquivos é composta pelo nome do serviço e de uma lista de máquinas, que podem (allow) ou não podem (deny) usar o serviço. Ao receber uma solicitação de uma máquina qualquer, o servidor faz o seguinte:

- Primeiro, verifica no arquivo *hosts.allow* se a máquina está incluída nele. Se estiver, seu acesso estará permitido.
- Caso contrário, o servidor verifica no arquivo *hosts.deny* se a máquina está incluída nele; se estiver, o acesso é negado.
- Se a máquina não estiver em nenhum dos dois arquivos, o acesso também é permitido.

Além de controlar os serviços inetd, tais como telnet e FTP, esses arquivos também podem ser utilizados para restringir a conexão aos daemons envolvidos no serviço NFS. O primeiro daemon a ter seu acesso restringido é o portmapper, que tem como função informar a porta em que cada serviço responde. Restringir o acesso ao portmapper é a melhor defesa contra qualquer um que queira invadir o seu sistema a partir do NFS, porque usuários não autorizados não poderão encontrar os daemons NFS. Todavia, há dois pontos a considerar:

- Restringir o acesso ao portmapper não é suficiente se algum intruso já souber, por alguma razão, como encontrar todos os daemons;

- Se o Network Information Service (NIS) está sendo executado, restringir portmapper também restringirá pedidos ao NIS. Em geral, é também aconselhável negar acesso a endereços IP aos quais não seja preciso permitir acesso explicitamente. O primeiro passo para fazer isso é adicionar a seguinte entrada ao arquivo */etc/hosts.deny*:

```
portmap:ALL
```

Adicione entradas para cada um dos daemons NFS:

```
lockd:ALL
mountd:ALL
rquotad:ALL
statd:ALL
```

- Restringir acesso a portmapper é a melhor defesa contra ataques a NFS.

- *hosts.deny*:

```
portmap:ALL, lockd:ALL, mountd:ALL, rquotad:ALL, statd:ALL
```

- *hosts.allow*.

```
service: host [ou network/netmask], host [ou network/netmask]
```

Alguns administradores de sistemas preferem colocar a entrada ALL:ALL no arquivo */etc/hosts.deny*, fazendo com que o acesso seja negado a todos os hosts, a menos que seja explicitamente permitido. Depois, é necessário adicionar uma entrada ao *hosts.allow* para disponibilizar o acesso aos hosts. As entradas em *hosts.allow* seguem o formato:

```
service: host [ou network/netmask], host [ou network/netmask]
```

Onde host é o endereço IP de um cliente e pode ser possível, em algumas versões, usar o nome DNS do host, embora não seja aconselhável. Suponha que desejamos permitir acesso a slave1.foo.com e slave2.foo.com e que os endereços IP dessas máquinas sejam 192.168.0.1 e 192.168.0.2, respectivamente. Teríamos, portanto, as seguintes entradas para */etc/hosts.allow*:

```
portmap: 192.168.0.1, 192.168.0.2
```

Para versões recentes do *nfs-utils*, pode-se adicionar:

```
lockd: 192.168.0.1, 192.168.0.2
rquotad: 192.168.0.1, 192.168.0.2
mountd: 192.168.0.1, 192.168.0.2
statd: 192.168.0.1, 192.168.0.2
```

## Iniciando os serviços NFS

Pré-requisitos:

- Versão do kernel que suporte NFS e versão adequada de *nfs-utils*.

Iniciando o portmapper.

- Localizado em */sbin* e, algumas vezes, em */usr/sbin*.

## Pré-requisitos

O servidor NFS deve ser configurado. Primeiro, é preciso ter os pacotes apropriados instalados. Isso consiste principalmente numa versão de kernel que suporte NFS e uma versão adequada do pacote `nfs-utils`.

Em seguida, antes de iniciar o NFS, verifique se a rede TCP/IP está funcionando corretamente na sua máquina. Feito isso, já é possível iniciar o NFS e executá-lo após reboot da máquina, pois os scripts de inicialização já detectam que o arquivo `/etc/exports` foi configurado e inicializará o NFS. Se não funcionar corretamente, alguns daemons precisarão ser inicializados para executar os serviços NFS.

## Iniciando o portmapper

NFS depende do daemon portmapper, também chamado portmap ou `rpc.portmap`. Ele deve estar localizado em `/sbin` e, algumas vezes, em `/usr/sbin`. Distribuições mais recentes do Linux inicializam esse daemon no processo de boot; de qualquer modo, é importante ter certeza de que ele está executando antes de trabalhar com o NFS. Digite:

```
ps aux | grep portmap
```

Daemons do pacote `nfs-utils`.

- **rpc.nfsd**: daemon do NFS que faz a maior parte do trabalho.
- **rpc.lockd** e **rpc.statd**: controlam os acessos de leitura e gravação de arquivos.
- **rpc.mountd**: manipula os pedidos iniciais de montagem de file systems.
- **rpc.rquotad**: gerencia as quotas de disco dos usuários em volumes exportados.

Lockd é chamado pelo nfsd sob demanda, portanto não se preocupe em iniciá-lo. Statd precisa ser iniciado separadamente. Distribuições Linux mais recentes têm scripts de inicialização para esses daemons.

Os daemons citados são parte do pacote `nfs-utils` e podem estar nos diretórios `/sbin` ou `/usr/sbin`. Se a distribuição em uso não inclui-los nos scripts de inicialização, então devemos adicioná-los e executar configuração para que iniciem na seguinte ordem:

1. `rpc.portmap`
2. `rpc.mountd`
3. `rpc.nfsd`, `rpc.statd`
4. `rpc.lockd` (se necessário)
5. `rpc.rquotad`

## Verificando a execução do NFS

- Consulte o portmapper com o comando `rpcinfo -p`.
- Se portmapper, nfs e mountd não estiverem listados, retorne e tente novamente inicializar os daemons.

- A saída é algo como:

```
program vers proto port
100000 2 tcp 111 portmapper
100000 2 udp 111 portmapper
...
```



Para fazer isso, consulte o portmapper com o comando *rpcinfo -p* para verificar os serviços que estão executando. A saída é algo como:

```
program ver protoport program ver proto port
100000 2 tcp 111 portmapper 100003 3 udp 2049 nfs
100000 2 udp 111 portmapper 100024 1 udp 944 status
100011 1 udp 749 rquotad 100024 1 tcp 946 status
100011 2 udp 749 rquotad 300019 1 tcp 830 amd
100005 1 udp 759 mountd 300019 1 udp 831 amd
100005 1 tcp 761 mountd 100021 1 udp 1042
nlockmgr
100005 3 tcp 771 mountd 100021 3 udp 1042 nlockmgr
100005 2 tcp 766 mountd 100021 4 udp 1042
nlockmgr
100005 3 udp 769 mountd 100021 1 tcp 1629
nlockmgr
100005 3 tcp 771 mountd 100021 3 tcp 1629
nlockmgr
100003 2 udp 2049 nfs 100021 4 tcp 1629
nlockmgr
```

Essa saída indica que as versões 2 e 3 do NFS, a versão 1 do *rpc.statd* e as versões 1, 3 e 4 do *rpc.lockd* estão funcionando. Há também outros serviços listados, dependendo do NFS estar sobre TCP ou UDP. O Linux usa UDP por default, a menos que TCP seja explicitamente requisitado.

Se não estiverem listados os serviços *portmapper*, *nfs* e *mountd*, então é necessário retornar e inicializar novamente os daemons.

## Configuração de cliente NFS

Montagem de diretórios remotos:

- Cliente NFS deve executar o *portmapper*.
- No cliente e no servidor são necessários os daemons:
  - ▣ *rpc.statd*.
  - ▣ *rpc.lockd*.





Exemplo:

```
# mount master.foo.com:/home /mnt/home
```

Antes de começar, deve-se ter certeza de que o cliente tem suporte para montagem NFS. Se estiver sendo usado o kernel com o sistema de arquivos */proc*, é possível verificar o arquivo */proc/filesystems* e ter certeza de que há uma linha contendo NFS. Se não, digite *insmod nfs* para ter certeza de que o NFS foi compilado como um módulo; caso contrário, é necessário construir ou fazer download de um kernel com suporte para NFS.

Para usar uma máquina como cliente NFS, é necessário que o portmapper esteja rodando nela. Para usar o controle de acesso de arquivos NFS, será necessário também que *rpc.statd* e *rpc.lockd* estejam rodando no cliente e no servidor. Distribuições mais recentes iniciam esses serviços por default, em tempo de boot.

Com portmap, lockd e statd executando, o administrador deve ser capaz de montar o diretório remoto de seu servidor da mesma forma que é montado um disco local, com o comando *mount*. Suponha que um dado servidor é chamado *master.foo.com* e que queremos montar o diretório */home* em *slave1.foo.com*. Basta digitar:

```
# mount master.foo.com:/home /mnt/home
```

O diretório */home* em master aparecerá como diretório */mnt/home* em slave 1 (do último exemplo).



Montagem em tempo de boot:

- O tipo de sistema de arquivos tem de ser NFS e dump e fsck devem ter valor zero.

Modos de falha:

- Soft:
  - ▣ Se um arquivo falha, o cliente NFS reportará erro.
  - ▣ Pode causar corrupção de arquivos e perda de dados.
  - ▣ Não se recomenda esse tipo de configuração.
- Hard:
  - ▣ Quando o servidor NFS voltar ao estado on-line, o programa continuará do ponto em que parou.
  - ▣ Recomenda-se esse tipo de configuração.

## Montagem de sistemas de arquivos NFS em tempo de boot

Sistemas de arquivos NFS podem ser adicionados no arquivo */etc/fstab* da mesma maneira que os sistemas de arquivos locais. A única diferença é que o tipo de sistema de arquivos será setado para NFS, e dump e fsck terão de ser ajustados para zero. Então, para o exemplo, a entrada no */etc/fstab* seria:

```
# device    mountpoint  fs-type  options  dump  fsck
...
master.foo.com:/home /mnt  nfs      rw       0    0
...
```

Se estiver sendo usado um automounter como *amd* ou *autofs*, as opções nos campos correspondentes das listagens do *mount* seriam similares, se não idênticas.





## Opções de montagem

Há várias opções a considerar em relação à forma pela qual o cliente NFS manipula uma falha (crash) no servidor ou queda da rede. Existem dois modos de falha distintos:

- **Soft:** se um arquivo falha, o cliente NFS reportará um erro para o processo na máquina cliente requisitando o acesso ao arquivo. Não se recomenda esse tipo de configuração, porque pode causar corrupção de arquivos e perda de dados;
- **Hard:** o processo não pode ser interrompido ou encerrado, a menos que isso seja especificado, após uma queda no servidor. Quando o servidor NFS voltar ao estado on-line, o programa continuará do ponto em que parou. Recomenda-se o uso dessa configuração.

Considerando o exemplo anterior, a entrada *fstab* seria:

```
# device      mountpoint fs-type  options  dump fsck
...
master.foo.com:/home /mnt/home nfs     rw,hard,intr 0 0
...
```

Tamanho do bloco:

- “*rsz*” e “*wsz*” especificam o tamanho.
  - Não há um tamanho que funcione bem em todos ou na maioria dos setups.
  - Escolher um tamanho de bloco correto é um fator importante para o desempenho.

### Configurando o tamanho de bloco para otimizar a velocidade de transferência

As opções de montagem “*rsz*” e “*wsz*” especificam o tamanho dos blocos de dados que o cliente e o servidor passam um para o outro.

Os valores default podem ser muito grandes ou muito pequenos; não há um tamanho que funcione bem em todos ou na maioria dos setups. Por outro lado, algumas combinações de kernels Linux e placas de rede não podem manipular blocos muito grandes. No entanto, se eles podem manipular blocos maiores, um tamanho maior pode ser mais rápido.



Escolher o tamanho de bloco correto é um fator importante na performance e é crucial se o servidor NFS tiver de ser usado em ambientes de produção.

### Otimizando o desempenho do NFS

- O desempenho do NFS pode ser aumentado com o ajuste de tamanhos dos buffers de transferência de dados.
- Valores adequados mudam de acordo com o caso.
- Recomendável fazer alguns testes para determinar esses valores.

Uma análise cuidadosa de seu ambiente, cliente e servidor, é o primeiro passo necessário para otimizar o desempenho do NFS.

Importantes configurações de otimização no lado cliente estão relacionadas aos tamanhos dos buffers de transferência de dados, especificados pelas opções “*rsz*” e “*wsz*” do comando *mount*.



Tamanho dos blocos:

- “rsize” e “wsize” do comando *mount*.

Exemplos de medição:

```
# time dd if=/dev/zero of=/mnt/home/testfile bs=16k count=16384
# time dd if=/mnt/home/testfile of=/dev/null bs=16k
```

Os valores mais adequados devem ser setados em */etc/fstab*.

As opções “rsize” e “wsize” do comando *mount* especificam o tamanho dos blocos de dados que o cliente e o servidor trocam um com o outro. Se as opções “rsize” e “wsize” não são especificadas, serão usados valores default de cada versão NFS. O default mais comum é 4 K.

Para o NFSv3, o limite é específico para o servidor. No servidor Linux, o tamanho máximo do bloco é definido pelo valor constante *NFSSVC\_MAXBLKSIZE*, encontrado no arquivo-fonte *./include/linux/nfsd/const.h*.

Os defaults podem ser grandes ou pequenos demais, dependendo da combinação de kernel e hardware. Por outro lado, algumas combinações de kernels do Linux de determinadas placas de rede não podem manipular blocos muito grandes. No entanto, se blocos maiores podem ser manipulados, um tamanho maior representa mais rapidez.

É recomendado experimentar e encontrar “rsize” e “wsize” que funcionem tão rápido quanto possível. Pode-se testar a velocidade de suas opções com alguns comandos simples, se o ambiente de rede não for tão pesado. Note que seus resultados podem variar a menos que sejam utilizados benchmarks mais complexos, como Bonnie, Bonnie++ ou IOzone.

O comando *dd* a seguir transfere 16 K para a partição montada. Para saber o tempo gasto, basta digitar:

```
# time dd if=/dev/zero of=/mnt/home/testfile bs=16k count=16384
```

Um arquivo de 256 Mb de bytes zerados é criado. Em geral, deve-se criar um arquivo que tenha, pelo menos, duas vezes o tamanho da memória RAM do servidor. Certifique-se de que existe espaço suficiente em disco. Agora, leia novamente o arquivo na máquina cliente a partir de */dev/null*, digitando:

```
# time dd if=/mnt/home/testfile of=/dev/null bs=16k
```

Repita o procedimento algumas vezes e obtenha uma média.

Para a determinação dos valores mais adequados, deve-se utilizar ferramentas de benchmark como:

- Bonnie.
- Bonnie++.
- IOzone file system benchmark.
- NFS benchmark.

O teste deve durar algumas horas e ser repetido para diferentes valores.

Em seguida, desmonte e monte novamente com um tamanho de bloco maior e menor. Eles devem ser múltiplos de 1.024 e não podem exceder o tamanho máximo do bloco permitido pelo sistema. O NFSv3 suportará até 64 K, se permitido. O tamanho do bloco deve ser uma potência de dois, já que a maioria dos parâmetros (como tamanhos de bloco do sistema e tamanho do pacote de rede) são potências de dois. Após montar com tamanho maior, acesse o sistema de arquivos montado e explore-o um pouco para ter certeza de que tudo está como esperado.



Lembre-se de editar */etc/fstab* para refletir o *rsize/wsize* mais adequado. Se os resultados parecem inconsistentes, ou suscitam dúvidas, pode ser necessário o uso de analisadores que executam testes com vários valores de “rsize” e “wsize”. O benchmark deve ser executado para cada valor de “rsize” e “wsize” que seja de interesse.

## Tamanho de pacote e drivers de rede

Comando *nfsstat*:

- Verifica as transações NFS, estatísticas de clientes, servidores e da rede etc.

Comando *tracepath*:

- Verifica o tamanho do pacote de rede.

*/proc/net/snmp*:

- Informações sobre o comportamento corrente da rede.

Para análise mais extensiva do comportamento NFS em particular, pode ser utilizado o comando *nfsstat* para verificar transações NFS, estatísticas de clientes e servidores, estatísticas da rede e assim por diante. A opção “-o net” mostrará o número de pacotes descartados em relação ao número total de transações.

Em transações UDP, a estatística mais importante é o número de retransmissões, devido aos pacotes descartados, overflows, congestionamentos, timeouts etc. Isso terá efeitos no desempenho do NFS e deve ser monitorado cuidadosamente. Note que *nfsstat* ainda não implementa a opção “-z”, que zera todos os contadores, então os valores dos contadores devem ser anotados antes de executar os benchmarks. Para resolver problemas de rede, pode-se reconfigurar o tamanho do pacote usado pelas placas de rede.

O tamanho do pacote de rede pode ser determinado com o comando *tracepath*: da máquina cliente, digite *tracepath server 2049* e o MTU do caminho deverá ser informado. Pode-se então ajustar o MTU da sua placa de rede igual ao MTU do caminho, usando a opção “MTU” para *ifconfig*, e verificar se há número menor de pacotes descartados.

Em adição, o comando *netstat -s* dará as estatísticas coletadas para o tráfego dos protocolos suportados. Em */proc/net/snmp* podem ser obtidas informações sobre o comportamento corrente da rede.

## Overflow de pacotes fragmentados

- “rsize” ou “wsize” maiores que o MTU da rede podem causar fragmentação de pacotes quando se usa NFS sobre UDP e quando a fragmentação requer recursos de CPU.
- Pacotes podem ser descartados.
  - Rotas de fragmentos diferentes.
  - Capacidade do servidor NFS.

A utilização de “rsize” ou “wsize” maior do que o MTU da sua rede causará fragmentação de pacotes IP quando usarmos NFS sobre UDP. A fragmentação e montagem de pacotes requer uma quantidade significativa de recursos de CPU. Em adição, a fragmentação de pacotes também expõe o tráfego na rede a riscos de não confiabilidade, já que um pedido RPC completo deve ser retransmitido se um pacote fragmentado UDP é descartado por alguma razão. Qualquer aumento de retransmissões RPC impacta na possibilidade de timeouts maiores, e isso é o pior impedimento em termos de desempenho para o NFS sobre UDP.

Pacotes podem ser descartados por várias razões. Se a topologia da sua rede é complexa, rotas de fragmentos podem ser diferentes, fazendo com que nem todos os fragmentos cheguem ao servidor para remontagem. A capacidade do servidor NFS pode ser outro fator de impacto, já que o kernel tem um limite de fragmentos que pode colocar em buffer antes de disparar os pacotes. Com kernels que suportam o sistema de arquivos */proc*, é possível monitorar os arquivos:

- */proc/sys/net/ipv4/ipfrag\_high\_thresh*
- */proc/sys/net/ipv4/ipfrag\_low\_thresh*

Uma vez que o número de pacotes fragmentados e não processados atinja o número especificado pelo *ipfrag\_high\_thresh* (em bytes), o kernel simplesmente iniciará o envio dos pacotes fragmentados até que o número de pacotes incompletos atinja o número especificado em *ipfrag\_low\_thresh*.

Outro contador para monitorar é o IP: ReasmFails no arquivo */proc/net/snmp*, representando o número de falhas de remontagem de fragmentos.

## NFS sobre TCP

Vantagens:

- Funciona melhor que UDP em determinadas redes.
- Melhor desempenho.
- Manipula melhor as diferenças de velocidade de rede.

Desvantagem:

- Não é um protocolo stateless como o UDP.

O overhead gerado pelo protocolo TCP resultará em menor performance do que UDP sobre condições ideais de rede.

O uso de TCP tem uma vantagem e uma desvantagem em relação à utilização de TCP sobre UDP.

- **Vantagem:** TCP funciona melhor do que UDP em determinadas redes. Quando se usa TCP, um pacote descartado pode ser retransmitido sem a retransmissão do pedido RPC inteiro, resultando em melhor desempenho. Em adição, TCP manipulará diferenças de velocidade de rede melhor do que UDP, devido ao fluxo de controle básico no nível de rede.
- **Desvantagem:** TCP não é um protocolo stateless como o UDP. Se o servidor parar no meio de uma transmissão de pacotes, os compartilhamentos serão desmontados e remontados. O overhead gerado pelo protocolo TCP resultará em menor performance do que UDP sob condições ideais de rede, mas o custo não é crítico.

Duas opções do comando *mount timeo* e *retrans* controlam o comportamento de pedidos UDP quando o cliente sofre timeouts devido aos pacotes descartados, congestionamento de rede e assim por diante.

- **Opção “-o timeo”:** permite a designação do tempo, em décimos de segundo, que levará até que o cliente decida que não vai esperar uma resposta do servidor e tentará enviar o pedido novamente. O valor default é de 7 décimos de segundo.



- **Opção “-o retrans”:** define o número de timeouts permitidos antes de o cliente desistir e exibir a mensagem de que o servidor não responde. O valor default é de três tentativas. Uma vez que o cliente mostra essa mensagem, ele continuará a tentar enviar o pedido, mas apenas uma vez antes de mostrar a mensagem de erro se outro timeout ocorrer. Quando o cliente restabelece contato, ele estará usando os valores corretos de *retrans* e mostrará uma mensagem “OK” no servidor.

A critério do administrador da rede, esses valores podem ser ajustados, caso esteja havendo quantidade excessiva de retransmissões ou se deseje aumentar o tamanho do bloco de transferência sem contar timeouts e retransmissões.

- Timeout e valores de retransmissão.
- *timeo* e *retrans* controlam o comportamento de pedidos UDP quando o cliente sofre timeouts.
- Número de instâncias de *nfsd*.
- A maioria dos scripts de inicialização disparam oito instâncias desse daemon.
- Pode ser modificado, via argumento, para */etc/rc.d/init.d/nfs* (no Red Hat).



A maioria dos scripts de inicialização iniciam oito instâncias de *nfsd*. Não há um número ótimo de instâncias, todavia, um servidor mais pesado pode requerer mais instâncias. Pode ser utilizado pelo menos um daemon por processador, mas de quatro a oito por processador pode ser melhor.

A utilização a partir do kernel 2.4 pode verificar como cada thread *nfsd* está sendo usada no arquivo */proc/net/rpc/nfsd*. Os últimos dez números daquele arquivo indicam o número de segundos de uso de thread em porcentagem do máximo permitido. Para aumentar o número de instâncias *nfsd*, inicie o *nfsd* usando o número de instâncias no argumento *RPCNFSDCOUNT* do script de inicialização NFS:

```
/etc/rc.d/init.d/nfs (no Red Hat)
/etc/init.d/nfs-kernel-server (no Debian)
```

## NFS síncrono versus assíncrono

Comportamento assíncrono de NFSv3.

- *async*.
- Servidor responde a pedido dos clientes assim que processado e manipulado pelo sistema de arquivos local.
- Não espera pelo armazenamento dos dados em meio estável.



O comportamento de NFSv3 usado pelo *exportfs* é assíncrono. Esse valor default permite que o servidor responda aos pedidos clientes tão brevemente quanto são processados e manipulados os pedidos pelo sistema de arquivos local, sem esperar que os dados sejam escritos em meios de armazenamento estável.

A opção “*async*” da lista de exportação do servidor pede ao servidor que informe ao cliente que todos os dados foram escritos para armazenamento estável, independentemente do protocolo usado. Para seguir um comportamento síncrono, o sistema de arquivos do servidor Linux deve ser exportado com a opção “*sync*”.

Note que especificar uma exportação síncrona resultará na situação em que nenhuma opção será vista na lista de exportação do servidor. Para exportar sistemas de arquivos, podem ser usadas diferentes opções:

```
# /usr/sbin/exportfs -o rw,sync */usr/local
# /usr/sbin/exportfs -o rw */tmp
```

Parâmetros do sistema de arquivos exportado:

```
# /usr/sbin/exportfs -v
/usr/local *(rw)
/tmp *(rw,async)
```

Exportação assíncrona:

- */proc/fs/nfs/exports*.
  - ▣ Mostra a lista de opções de exportação.
- Clientes podem explicitamente insistir em comportamento síncrono total, independente de protocolo, usando a opção “O -SYNC” ao abrir arquivos.

Se o kernel for compilado com o sistema de arquivos */proc*, o arquivo */proc/fs/nfs/exports* também mostrará a lista completa de opções de exportação.

Quando o comportamento síncrono é especificado, o servidor não concluirá um pedido NFSv2 até que o sistema de arquivos local tenha escrito todos os dados no disco. O servidor concluirá um pedido assíncrono NFSv3 sem esse delay e retornará o status dos dados para informar ao cliente os dados que devem ser mantidos em cache e os dados que podem ser descartados de maneira segura. Há três valores possíveis de status, definidos em *include/linux/nfs.h*:

- **NFS\_UNSTABLE**: os dados não foram armazenados e confirmados em armazenamento estável no servidor e devem ser colocados em cache no cliente até que um pedido de confirmação do cliente garanta que o servidor enviou os dados para armazenamento estável;
- **NFS\_DATA\_SYNC**: os meta dados não foram enviados para armazenamento estável, e devem ser colocados em cache no cliente. Uma confirmação subsequente é necessária;
- **NFS\_FILE\_SYNC**: dados e meta dados não podem ser colocados em cache, e uma confirmação subsequente não precisa ser enviada para a faixa coberta por esse pedido.

Em adição ao comportamento síncrono, o cliente pode explicitamente insistir em comportamento síncrono total, independente do protocolo, abrindo todos os arquivos com a opção O-SYNC. Nesse caso, todas as respostas aos pedidos dos clientes vão esperar até que os dados atinjam o disco do servidor.

## Segurança e NFS

Passos para que um cliente possa acessar um arquivo num diretório remoto:

- Montagem: segurança provida por */etc/exports*.
- Acesso aos arquivos: não é específico do NFS, mas parte do controle de acesso do sistema de arquivos no cliente.

Com NFS, há dois passos requeridos para que um cliente ganhe acesso a um arquivo contido em um diretório remoto no servidor.

O primeiro passo é o de montagem. O acesso à montagem é conseguido através da máquina cliente tentando atingir o servidor. Para tal, a segurança é provida pelo arquivo `/etc/exports`, que lista os nomes ou endereços IP para máquinas que possam acessar o ponto de compartilhamento. Se alguém for capaz de efetuar *spoofing*, por exemplo, poderá acessar os pontos de montagem. Uma vez que a máquina foi montada num dado volume, seu Sistema Operacional terá acesso a todos os arquivos naquele volume, e também acesso de escrita se o volume tiver sido exportado com a opção `"rw"`.

O segundo passo é o acesso aos arquivos, uma função de controle de acesso do sistema de arquivos no cliente e não uma função especializada do NFS. Uma vez que um drive é montado, as permissões do usuário e do grupo nos arquivos determinam controle de acesso.

## NFS e segurança: portmapper

O portmapper mantém uma lista dos serviços em execução em determinadas portas.

- Algumas distribuições do Linux não se preocupam com a segurança do portmapper.

Como tornar mais seguro?

- Editar `/etc/hosts.deny` e incluir a linha:
  - ▣ `portmap:ALL`



O portmapper mantém uma lista dos serviços em execução em determinadas portas. Essa lista é usada por uma máquina conectada para ver as portas que ela deseja acessar para obter determinados serviços.

Algumas distribuições do Linux não se preocupam com a segurança do portmapper; com isso, a maneira mais fácil de fazer essa verificação é executando strings e verificando se foram lidos arquivos relevantes, como `/etc/hosts.deny` e `/etc/hosts.allow`. Supondo que o portmapper seja `/sbin/portmap`, pode-se fazer a verificação com o comando:

```
# strings /sbin/portmap | grep hosts
```

Numa máquina segura, a saída seria algo como:

```
/etc/hosts.allow
/etc/hosts.deny
@(#) hosts_ctl.c 1.4 94/12/28 17:42:27
@(#) hosts_access.c 1.21 97/02/12 02:13:22
```

Primeiro, editamos `/etc/hosts.deny`, que deve conter a linha `portmap: ALL`, que negará o acesso a todos. Após isso, execute `rpcinfo -p` apenas para verificar se o portmapper realmente está lendo o arquivo `/etc/hosts.deny`. Os arquivos `/etc/hosts.allow` e `/etc/hosts.deny` produzem efeito imediatamente após salvos. Não há necessidade de inicializar daemons.

Fechar portmapper para todo mundo é uma medida drástica, então podemos abri-lo novamente editando `/etc/hosts.allow`, listando todas as máquinas que devem acessar serviços na máquina.

## NFS para MS Windows

- Existem versões exclusivas do servidor NFS para o Windows, como o ProNFS.
- É possível criar um servidor NFS ou acessar qualquer cliente NFS.



Existem versões exclusivas do servidor NFS para o Windows, como o ProNFS. Com esse programa é possível criar um servidor NFS ou acessar qualquer cliente NFS. A diferença é que não é preciso necessariamente montar o diretório exportado no computador e acessar os clientes pelo ambiente de rede do Windows.

Ao instalar o ProNFS como padrão, ele não vai instalar a versão cliente. Para efetuar a instalação do cliente, selecione a opção “Custom” durante o processo e, no menu de seleção, clique no campo *NFS Client for NFS Network*. Após reiniciar o computador, verifique se o servidor NFS pode ser encontrado pelo utilitário NFSProbe, na aba “Programas” do menu “Iniciar”.







# Roteiro de Atividades 3

Nestas atividades, os alunos devem trabalhar com duas máquinas virtuais (servidor e cliente). Ambas devem estar na mesma rede. Como regra, o endereço 192.168.X.1 será o do servidor Linux e o endereço 192.168.X.2 será do cliente Linux. Teste o funcionamento da rede através do comando *ping*.

## Atividade 3.1 – Servidor de DNS Primário

---

Considerando a rede 192.168.X.0/255.255.255.0, cujo domínio é empresa.com.br, configure o servidor de DNS Primário de modo que ele tenha as seguintes máquinas registradas:

- servidor.empresa.com.br: 192.168.X.1 (NS)
- email.empresa.com.br: 192.168.X.4 (MX)
- cliente.empresa.com.br: 192.168.X.2
- windows.empresa.com.br: 192.168.X.3
- www.empresa.com.br: 192.168.X.1 (CNAME)
- pop.empresa.com.br: 192.168.X.4 (CNAME)
- smtp.empresa.com.br: 192.168.X.4 (CNAME)



Não se esqueça de configurar a resolução de nomes reversa.

## Atividade 3.2 – Servidor de DNS Secundário

---

Configurar o servidor de DNS Secundário na máquina virtual cliente (192.168.X.2). Lembre-se de informar o endereço do servidor secundário no parâmetro *allow-transfer* do Servidor Primário. Os arquivos de zona que forem transferidos devem ser gravados no diretório */etc/bind/sec* do servidor secundário.

## Atividade 3.3 – Exportando diretório do servidor NFS

---

Crie e exporte o diretório */dados* para a máquina cliente (192.168.X.2).

## Atividade 3.4 – Configuração do Cliente NFS

---

Instale o cliente NFS na máquina cliente (192.168.X.2), monte o diretório */dados* do servidor no diretório */mnt/remoto* e realize as configurações necessárias para que sempre que a máquina cliente for reiniciada o diretório */dados* seja montado automaticamente.

## Atividade 3.5 – Testando o funcionamento do serviço NFS

---

No servidor, crie um arquivo chamado *teste* no diretório */dados* e verifique se este aparece no cliente. Depois edite o arquivo *teste*, pelo cliente, adicionando a data atual ao conteúdo do arquivo, volte ao servidor e verifique se o arquivo foi alterado.





# 4

## Servidor LDAP

### objetivos

Entender o funcionamento do serviço de diretório, do protocolo LDAP e da solução OpenLDAP; compreender o funcionamento, objetos, atributos e instalação do LDAP; instalar OpenLDAP, com execução do servidor e criação e manutenção de base de dados.

Servidor LDAP, uso do script Migration tools para exportação de contas e grupos da máquina, inclusão de usuários e troca de senhas com PAM.

### conceitos

### Introdução

Conheceremos os conceitos de serviço de diretório e o protocolo Lightweight Directory Access Protocol (LDAP), além da solução OpenLDAP, em aspectos relacionados à instalação, configuração e testes.

Na primeira parte, abordaremos o funcionamento, objetos e atributos do protocolo LDAP. Na etapa seguinte será abordada a instalação da implementação OpenLDAP, e em seguida o arquivo de configuração e suas diferentes formas.

A execução do servidor, bem como a criação e a manutenção da base de dados, serão os objetos do final deste capítulo, onde serão colocados em prática os conceitos adquiridos.

### LDAP

- Um diretório é similar a uma base de dados.
- Pode replicar informações e reduzir tempo de resposta.
- Há diferentes maneiras de prover um serviço de diretórios.
- Métodos distintos permitem armazenamentos diferentes de informação.
- Protocolo baseado no acesso a serviços de diretórios X.500, executa sobre protocolos de transporte orientados para conexão.



Um diretório é uma base de dados especializada para leitura e busca das informações nela contidas. Diferentemente de um banco de dados tradicional, em que existe grande volume de operações de escrita e de leitura, no serviço de diretório as operações são predominantemente de leitura. Além disso, os diretórios podem ter habilidade para replicar informações, obtendo confiabilidade e disponibilidade enquanto podem reduzir o tempo de resposta.



Há diferentes maneiras de prover um serviço de diretórios. Métodos diferentes permitem diferentes tipos de informações a serem armazenadas no diretório, impõem diferentes requisitos na forma como as informações podem ser referenciadas, consultadas, atualizadas etc.

LDAP é um protocolo cliente/servidor para acessar serviços de diretórios, especificamente serviços baseados em X.500. LDAP executa sobre TCP/IP ou outros serviços de transferência orientados para conexão.



A especificação do protocolo pode ser encontrada na RFC 4511.

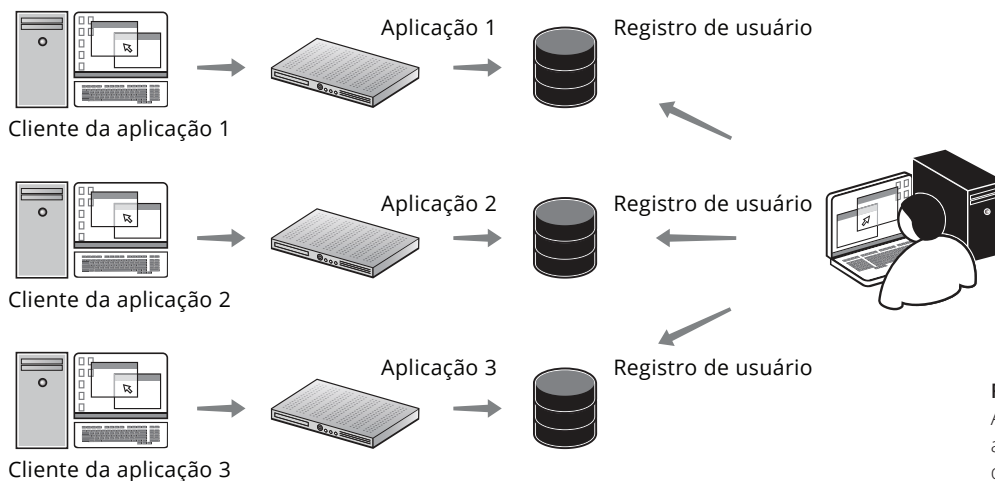
As informações em uma base LDAP são organizadas hierarquicamente, na forma de uma árvore. Essa estrutura representa geralmente a organização geográfica ou hierárquica de uma determinada organização. A partir de uma raiz, pode-se deslocar pela árvore até chegar a cada unidade organizacional, departamento, função, pessoa, equipamento etc. Cada entrada, conforme será explicado adiante, é identificada por nome único denominado Distinguished Name (DN).

## Sem serviço de diretório

Um serviço de diretório é fundamental para estruturas de sistemas distribuídos, pois é acessível a todas as aplicações e provê concepção simples de usuários, recursos e outros objetos necessários.

Quando o serviço de diretório está ausente em uma rede, cada aplicação possui sua própria base de dados para autenticar os clientes, cada uma com uma política de segurança diferente, de complexidade e tempos de expiração distintos, impondo ao usuário a necessidade de memorizar diferentes senhas.

Um novo usuário na rede demandará a criação de contas de acesso e senha em cada um dos sistemas. Do mesmo modo, a remoção de um usuário exigirá a sua exclusão em cada uma das bases. Assim, a ausência de um serviço de diretório implica em perda de produtividade e agilidade no cadastramento de usuários, além de possibilitar brechas de segurança ao manter usuários com acesso a determinados sistemas na rede, mesmo que por curto período de tempo. Em resumo, a administração se torna mais complexa.



**Figura 4.1**  
Aplicações  
acessando bases  
de dados diversas.

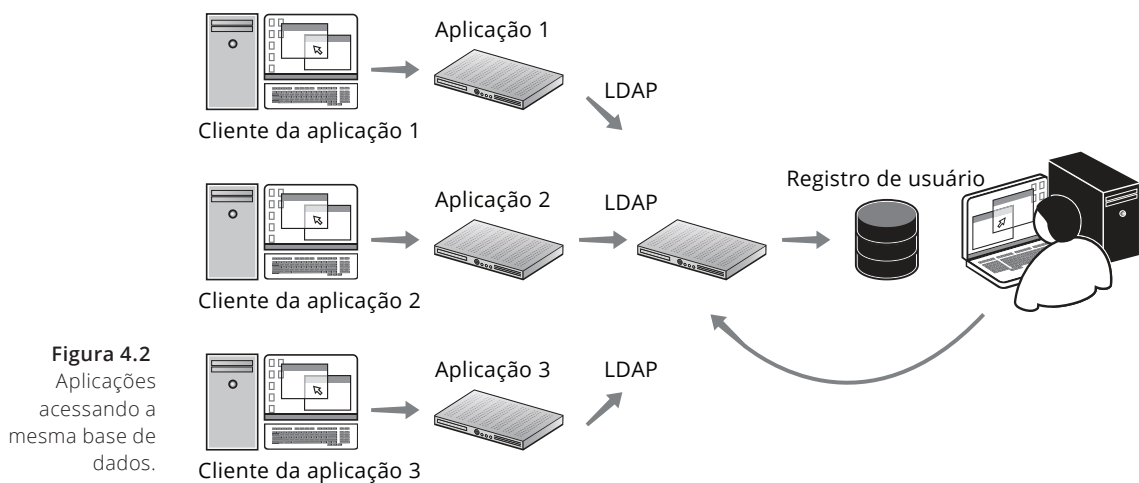
## Com serviço de diretório

A utilização de serviço de diretório possibilita a manutenção de base de dados única. Provê abstração lógica para uso por usuários, aplicações e outros recursos na rede, o que permite a usuários acessar os serviços de rede de maneira mais transparente.

O recurso mais comum de utilização de serviço de diretório está associado a catálogos de endereço, o que pode manter informações como: nome, e-mail, números de telefone, cargo, empresa etc.

Esse cadastro pode ser utilizado por quaisquer tipos de aplicações que suportem comunicação no padrão LDAP, tais como: sistemas para controle de acesso físico às instalações da empresa; aplicações de correio eletrônico e groupware; aplicações disponíveis na intranet ou internet; sistemas de folha de pagamento e financeiros.

Por exemplo: a manutenção de um cadastro único com todas as informações sobre os funcionários da empresa, podendo englobar desde os dados cadastrais até informações sobre as permissões de acesso a cada sistema e área da empresa. Desse modo, um funcionário admitido recentemente pode ter seus dados inseridos em um único local e ser visto imediatamente por todos os sistemas que dele necessitarem. Ou, no caso inverso, um funcionário demitido ou em férias pode ter seus privilégios restringidos imediatamente.



## Funcionamento do LDAP

Baseado no modelo cliente/servidor:

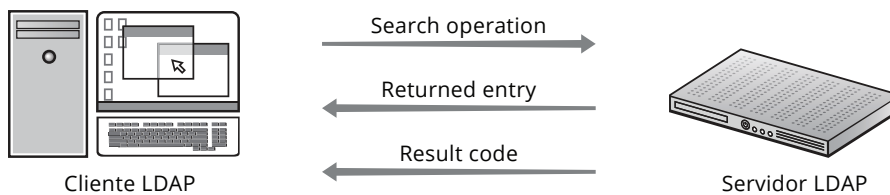
- Cliente requisita ao servidor.
- Servidor responde ou aponta para outro servidor LDAP.
- Tudo sob uma única visão.

O serviço de diretórios LDAP é baseado no modelo cliente/servidor. Um cliente LDAP conecta-se a um servidor LDAP e efetua requisições. O servidor responde ou então aponta para onde o cliente pode coletar mais informações (por exemplo, outro servidor LDAP). Independente de qual servidor LDAP um cliente se conecta, tudo deve apresentar a mesma visão de diretório, isto é, um nome apresentado por um servidor LDAP referencia a mesma entrada em outro servidor LDAP.

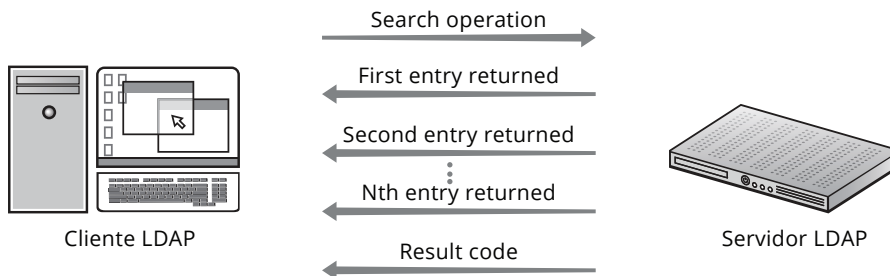
Operações:

- Consulta.
- Atualização.
- Autenticação e controle.

As RFCs 4510, 4511 e 4513 incluem definições sobre as especificações, formatos, esquemas e autenticação, entre outras informações.



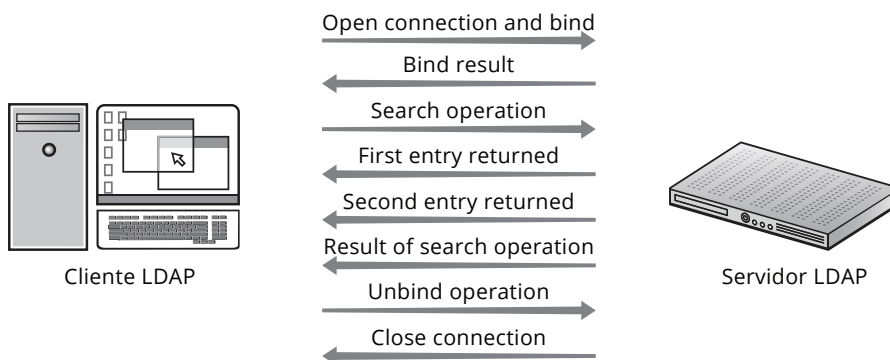
**Figura 4.3**  
Operação de busca simples.



**Figura 4.4**  
Operação de busca com múltiplos resultados.

O protocolo LDAP consiste em nove operações distribuídas em três categorias:

1. Operações de consulta:
  - **search**: operação de busca por um (Figura 4.3) ou múltiplos resultados (Figura 4.4);
  - **compare**: operação de comparação com determinada entrada.
2. Operações de atualização:
  - **add**: operação de adição de uma ou mais entradas na base;
  - **delete**: operação de remoção de dados da base;
  - **modify**: operação de modificação de dados da base;
  - **modify DN (rename)**: operação de modificação da posição de dados na base.
3. Operações de autenticação e controle:
  - **bind**: operação de autenticação;
  - **unbind**: operação de cancelamento da autenticação;
  - **abandon**: operação de cancelamento unilateral da consulta.



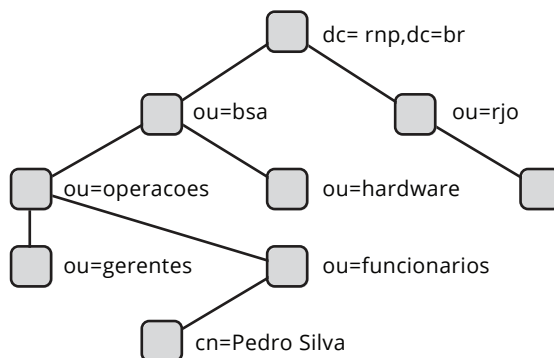
**Figura 4.5**  
Operação LDAP típica.

Em uma operação LDAP típica, é necessário um conjunto de passos para permitir o estabelecimento da conexão, autenticação, consulta, retorno dos dados e encerramento da conexão:

- Estabelecimento da conexão e solicitação de autenticação: open connection and bind.
- Resultado da autenticação: bind result.

- Operação de consulta: search operation.
- Retorno de um ou múltiplos resultados: entries return.
- Solicitação de cancelamento da autenticação: unbind operation.
- Encerramento da conexão: close connection.

## Organização dos dados



**Figura 4.6**  
Estrutura do  
diretório.

A base de dados LDAP pode ser representada na forma de uma árvore que representa os objetos de uma organização (unidade organizacional, departamento, função, pessoa, equipamentos, entre outros).

Cada entrada (objeto) de um diretório possui um nome distinto DN, que é uma forma de identificar cada entrada de forma única no diretório. Uma entrada é composta por um conjunto de atributos, e cada atributo descreve uma característica particular do objeto. Cada atributo tem um tipo e um ou mais valores. O tipo descreve a espécie de informação contida nos atributos, e o valor é o próprio dado.

No exemplo apresentado, consta uma entrada Pedro Silva, um funcionário do departamento de operações da filial BSA da organização RNP. Essa organização da árvore é descrita na RFC no item "Naming model", que descreve a organização e a forma de se referir aos dados.

## Exercício de fixação 1

### Estrutura de diretório LDAP

Monte uma estrutura de diretório para uma empresa de TI cujo domínio é empresa.com.br.

- Possui os serviços de Samba, Proxy, E-mail e Intranet.
- A empresa possui funcionário CLT e terceirizados.
- Todos têm acesso aos serviços Samba e Proxy.
- Somente os CLTs possuem acesso a e-mail e intranet.

## Tipos de dados e unidade básica de informação

| Atributos |         |
|-----------|---------|
| Tipo      | Valores |
| cn        | Aluno1  |
| sn        | ESR     |



| Atributos       |                    |
|-----------------|--------------------|
| uid             | aluno1             |
| telephoneNumber | +55 61 32434300    |
| mail            | pedro.silva@rnp.br |

**Tabela 4.1**  
Uma entrada com os atributos.

As entradas em uma base de dados podem ser apresentadas no formato LDAP Data Interchange Format (LDIF), utilizado para a importação ou exportação de dados a partir do servidor LDAP.

Exemplo de entrada no formato LDIF e seus atributos:

dn: distinguished name

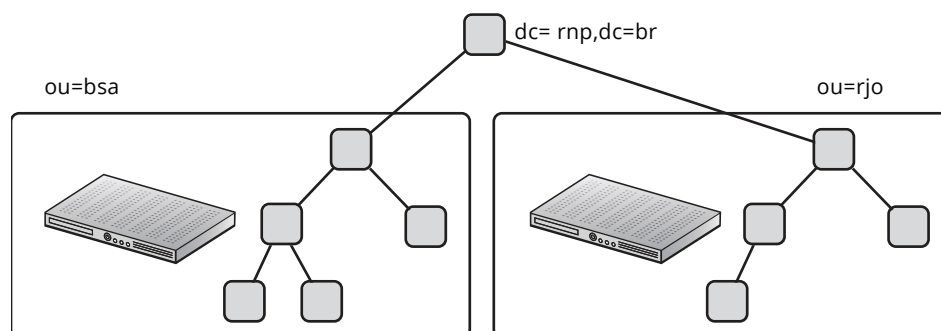
dc: domain component ou organizational unit

cn: common name

sn: surname

```
dn: uid=aluno1, ou= operacoes, dc= rnp, dc= br
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
cn: Aluno1 ESR
givenName: Aluno1
sn: ESR
uid: aluno1
mail: aluno1@ucb.br
Telephonenumber: +55 61 32434300
```

Os tipos de dados e a representação da unidade básica de informação são descritos na RFC 4512, que trata dos modelos de informação (Information Models).

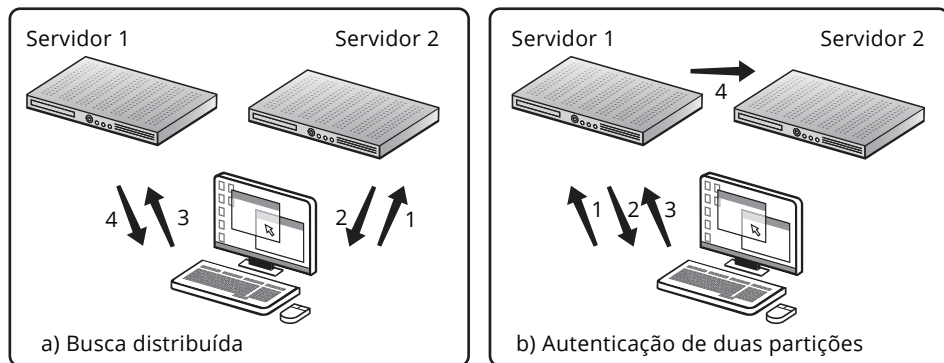


**Figura 4.7**  
Diretório distribuído.

A utilização de um serviço de diretório normalmente tem como objetivo a centralização da informação em uma única base de dados. No entanto, isso não quer dizer que esses dados devem ficar centralizados fisicamente em um único lugar.

Empresas ou organizações que possuem suas instalações dispersas geograficamente em múltiplos locais podem, por questões administrativas ou de desempenho, manter os dados de cada localidade localmente, embora todos agrupados em uma única base de dados. Desse modo, as informações acessadas mais frequentemente por determinada localidade ficam disponíveis localmente. Caso haja necessidade de acessar dados mantidos por outras filiais, isso pode ser feito de modo transparente.

### Referência e autenticação entre vários servidores



**Figura 4.8**  
Relacionamento  
entre servidores.

Um servidor pode manter somente parte da árvore, estando o restante dela em outros servidores. Nesse caso, cada servidor possui uma referência sobre o local onde estão as outras partes da árvore. Ao receber uma consulta sobre determinados dados que estejam em outro servidor, o servidor que recebeu a solicitação repassa a referência ao cliente que a solicitou (Figura 4.8a):

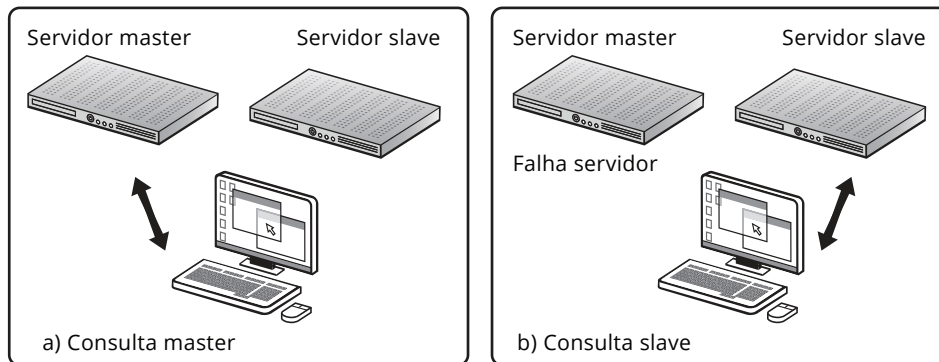
1. Cliente envia pedido de busca para o Servidor 2;
2. Servidor 2 retorna referência (referral) do Servidor 1 ao cliente;
3. Cliente envia pedido de busca para o Servidor 1;
4. Servidor 1 retorna as entradas associadas à busca.

Outra associação comum diz respeito à relação de confiança entre servidores LDAP. Nesse caso, um cliente autenticado em um determinado servidor pode ter sua solicitação encaminhada a outro servidor sem necessidade de nova autenticação (Figura 4.8b):

1. Cliente conecta e autentica no Servidor 1;
2. Servidor retorna código de sucesso ao cliente;
3. Cliente envia pedido de busca para o Servidor 1;
4. Servidor encaminha pedido para o Servidor 2.

Uma variação desse caso é o Servidor 1 informar ao Servidor 2 que o cliente já está autenticado, repassando a consulta para que o Servidor 2 a responda diretamente.

## Replicação



**Figura 4.9**  
Consulta simples.

A utilização de serviço de diretório possibilita ganhos em vários aspectos no ambiente de rede, mas torna o ambiente dependente do serviço, uma vez que sua indisponibilidade impede que usuários efetuem login em suas máquinas ou acessem sua lista de contatos, entre outros.

Em virtude disso, em muitos casos é necessária a utilização de um servidor principal (master) acompanhado de um ou mais servidores secundários (slaves).

Desse modo, a indisponibilidade de um servidor ou da rede, caso este esteja em outro segmento de rede, não impedirá o funcionamento do serviço de diretório (Figura 4.9).



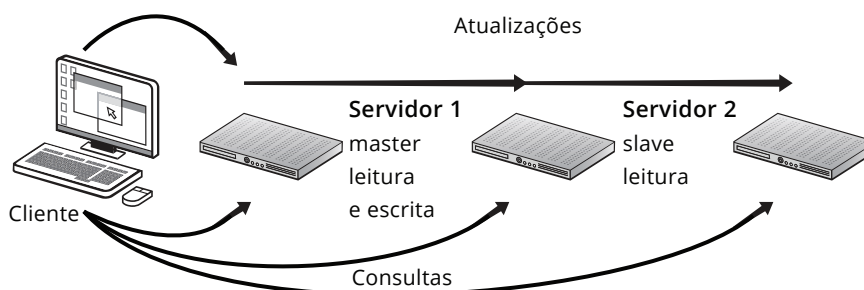
No servidor primário estará em execução o daemon SLAPD, enquanto que em cada servidor secundário estará em execução o daemon slurpd.

As configurações a serem utilizadas pelo SLAPD serão lidas de um arquivo denominado `/usr/local/etc/openldap/slapd.conf` ou de uma base LDAP localizada em `/usr/local/etc/openldap/slapd.d`.

A primeira opção (com `slapd.conf`) é mais amplamente utilizada e deverá ser usada por enquanto, principalmente se o serviço de diretório estiver sendo gerenciado por algum aplicativo específico designado para este fim (backend). A desvantagem dessa abordagem é a necessidade de reiniciar o servidor quando modificações forem realizadas no arquivo de configuração.

A segunda opção, que utiliza uma base de dados LDAP para armazenar as configurações do servidor, não necessita da reinicialização do servidor a cada modificação no arquivo de configuração.

### Modificações de cliente



**Figura 4.10**  
Consulta a múltiplos servidores.

Utilizando uma solução que contemple vários servidores, cada um deles pode atender às operações de consulta (mais frequentes), enquanto somente o servidor master pode efetuar operações de escrita na base de dados. As modificações são replicadas para os servidores slaves (Figura 4.10).

O mecanismo da replicação envolve as seguintes etapas:

1. O cliente LDAP envia uma operação de modificação para o servidor slave;
2. O servidor slave retorna uma referência (referral) do servidor master ao cliente;
3. O cliente LDAP envia a operação de modificação ao servidor master;
4. O servidor master efetua a operação, atualiza o arquivo de log e envia mensagem de sucesso ao cliente;
5. O daemon slurpd, presente no servidor master, percebe a operação pela modificação do arquivo de log e notifica o servidor slave via LDAP;
6. O servidor slave efetua a modificação e envia mensagem de sucesso ao daemon slurpd.

## Objetos e atributos LDAP

SLAPD provê suporte a diferentes bases de dados:

- BDB.
- LDBM.
- SHELL.
- PASSWD.

BDB é indicado para aplicações que requerem:

- Transações.
- Habilidade na recuperação de crashes e falhas.

O daemon servidor LDAP, denominado SLAPD, provê suporte a diferentes bases de dados:

- **Berkeley DataBase (BDB)**: base transacional de alta performance;
- **DataBase Manager (LDBM)**: base mais leve baseada no DBM;
- **SHELL**: interface para scripts shell;
- **PASSWD**: interface para o arquivo */etc/passwd*.

BDB é indicado para acesso de leitura/escrita de bases de dados, com uma mistura de operações de leitura e escrita. É usado em aplicações que requerem:

- Transações, incluindo mudanças na base de dados e desfazendo alterações não confirmadas;
- Habilidade de recuperação de crashes e falhas de hardware sem perda das transações confirmadas.

## Formato LDIF

- Importa e exporta informações de diretório entre servidores de diretórios baseados em LDAP.
- Descreve um conjunto de mudanças aplicadas a um diretório.
- Armazena informações em hierarquia orientada a objetos de entrada.



Para importar e exportar informações de diretório entre servidores de diretórios baseados em LDAP ou para descrever um conjunto de mudanças aplicadas a um diretório, o formato de arquivo LDIF para transferência de dados LDAP é tipicamente usado. Um arquivo LDIF armazena informações numa hierarquia orientada a objetos de entrada. O pacote LDAP já vem com um utilitário para converter arquivos LDIF para o formato BDB.

No LDAP, uma classe objeto define a coleção de atributos que pode ser usada para definir uma entrada. O padrão LDAP provê os seguintes tipos de classes:

- Grupos no diretório, incluindo listas não ordenadas de objetos ou grupos de objetos;
- Localizações, tais como nome do país e descrição;
- Organizações no diretório;
- Pessoas no diretório.

## Classes LDAP

- Grupos no diretório, incluindo listas não ordenadas de objetos ou grupos de objetos.
- Localizações como nome do país e descrição.
- Organizações no diretório.
- Pessoas no diretório.
- Existem classes para descrever quase todos os tipos de objetos de que se necessita.
  - Classes atuais podem ser estendidas, se necessário.

Uma entrada pode pertencer a mais de uma classe de objetos. A estrutura das classes do servidor determina a lista total de atributos requeridos e permitidos para uma entrada particular. Por exemplo, a entrada para uma pessoa é definida pela classe *person*, mas também pode ser definida por atributos em *inetOrgPerson*, *groupOfNames*.

Qualquer pedaço específico de informação é associado com um atributo descritivo. Por exemplo, *cn* ou *commonName* é um atributo usado para armazenar um nome de usuário. Um usuário identificado como Aluno2 ESR pode ser representado no diretório como:

```
cn: Aluno2 ESR
```

Cada pessoa do diretório é definida por uma coleção de atributos da classe *person*. Outros atributos usados nessa classe incluem:

```
givenname: Aluno2  
surname: ESR  
mail: aluno2.esr@rnp.br
```

O atributo *ObjectClass* lista a classe a qual uma entrada pertence.

## Instalação do servidor LDAP

Passos para a instalação do servidor LDAP:

- Instalar os pacotes pré-requeridos (se ainda não estiverem instalados).
- Fazer download do servidor.
- Desempacotar o software.
- Configurar os makefiles.
- Construir os servidores.



Informações sobre as classes de objetos e os atributos associados podem ser conferidas na RFC 4519 – Lightweight Directory Access Protocol (LDAP): Schema for User Applications.





Pré-requisitos:

- Bibliotecas OpenSSL (SSL/TLS) e Kerberos.
  - ▣ BDB requer Sleepycat Software Berkeley DB v.4.
  - ▣ OpenLDAP suporta pthreads POSIX e Mach Cthreads, entre outras.

Distribuições do LDAP:

- Servidor da Universidade de Michigan.
- Servidor OpenLDAP.

Configuração do software:

- Fontes OpenLDAP são distribuídas com um script de configuração.

Requisitos necessários para a instalação do LDAP:

- **Bibliotecas OpenSSL (SSL/TLS):** são parte do sistema base LDAP;
- **Serviços de autenticação Kerberos:** clientes e servidores LDAP suportam serviços de autenticação baseados em Kerberos;
- **OpenLDAP:** suporta autenticação SASL/GSSAPI;
- **Base de dados:** Berkeley Data Base (BDB) requer Sleepycat Software Berkeley DB;
- **Threads:** LDAP é projetado para dispor as vantagens de threads; OpenLDAP suporta pthreads POSIX, Mach Cthreads, entre outras;
- **TCP Wrappers:** SLAPD suporta TCP Wrappers se estiver pré-instalado.

Passos necessários:

- Fazer download do pacote.
- Prefira a última versão estável (stable release).
- Desempacotar o software.



## Download do pacote

Existem algumas implementações de servidores LDAP:

- Servidor LDAP, da Universidade de Michigan (em geral é a base das outras soluções);
- Netscape Directory Server;
- OpenLDAP;
- Fedora Directory Server;
- Sun Directory Server.

Entre as versões livres consideradas prontas e maduras, a mais utilizada é o OpenLDAP.

O servidor OpenLDAP é baseado na última versão do servidor da Universidade de Michigan. Há listas de discussão e documentação disponíveis. Nas atividades deste Capítulo será utilizado o OpenLDAP.

## Desempacotando o software

Para desempacotar o pacote LDAP, podemos usar a combinação:

```
# gunzip openldap-VERSION.tgz | tar xvf -  
  
# cd openldap-VERSION
```



! Substituir VERSION pelo número da versão a ser utilizada.

Passos necessários:

- Configurar makefiles com o comando:

```
./configure -- [parâmetros]
```

- Construir os servidores:

- `make depend`: constrói as dependências.
- `make`: constrói o servidor.
- `make test`: executa testes.
- `make install`: instala binários e manual.



## Configuração do software

As fontes do servidor OpenLDAP são distribuídas com um script para configurar opções como diretórios de instalação, compilador e flags de linker.

No diretório onde foi desempacotado o software, use o comando:

```
# ./configure -- help
```

Serão apresentadas todas as opções que podem ser customizadas com o script de configuração antes de construir o software.

É possível especificar algumas variáveis de ambiente com o comando `env` antes de executar o script de configuração:

- **CC**: especifica um compilador C alternativo;
- **CFLAGS**: especifica flags adicionais do compilador;
- **CPPFLAGS**: especifica flags do pré-processador C;
- **LDFLAGS**: especifica flags de linker;
- **LIBS**: especifica bibliotecas adicionais.

## Construindo o servidor

Após configurar o software, construa as dependências usando o comando:

```
# make depend
```

Depois, construa o servidor usando o comando:

```
# make
```

Para garantir construção correta, deve ser feito teste com:

```
# make test
```

Instale os binários e as páginas de manual:

```
# make install
```

# Configuração do servidor LDAP

## Arquivo */etc/slapd.conf*

- Todas as configurações do SLAPD são conseguidas por meio do arquivo */etc/slapd.conf*.
- Diretivas do arquivo de configuração:
  - **Global:** aplicadas a todos os backends e databases, exceto no caso de sobreposição.
  - **Backend specific:** aplicadas a todas as instâncias de bases de dados do mesmo tipo.
  - **Database specific:** marca o início da definição de uma nova instância de base de dados



Todas as configurações do SLAPD são obtidas por meio do arquivo */etc/slapd.conf*. As diretivas do arquivo de configuração são separadas em global, backend specific e database specific.

## Formato do arquivo de configuração

O arquivo *slapd.conf* consiste em três tipos de informação: global, backend specific e database specific. Diretivas globais podem ser sobrepostas por diretivas backend e database.

A distribuição do OpenLDAP contém um arquivo de configuração-exemplo que será instalado em */etc/openldap*. Definições de esquemas (tipos de atributos e classes de objetos) estão disponíveis em */etc/openldap/schema*.

## Diretivas globais

Diretivas globais se aplicam a todos os backends e databases, a menos que haja sobreposição na definição das respectivas diretivas. Por exemplo:

```
access to <what> [by <who><accesslevel><control>]
```

(garante acesso, especificado por *accesslevel*, a um conjunto de entradas e/ou atributos, especificados por *<what>*, por um ou mais requisitantes especificados por *<who>*).

```
attributetype<RFC4517 Syntaxes and Matching Rules>
```

Define um tipo de atributo. Por exemplo:

```
access to * by * read (garante a leitura para todos)

defaultaccess {none | compare | search | read | write}
```

(especifica o acesso default aos requisitantes quando nenhuma diretiva de acesso tiver sido especificada).

```
include <filename>
```

(especifica que SLAPD deve ler informações de configuração adicionais de determinado arquivo antes de continuar com a próxima linha do arquivo corrente).

## Diretivas backend

Diretivas aplicadas a todas as instâncias de bases de dados do mesmo tipo e, dependendo da diretiva, pode ser sobrescrita por diretivas database.

*backend <type>* (marca o início de uma definição backend).

Type pode ser um dos seguintes:

- Berkeley DB: *bdb*.
- DNS SRV: *dnssrv*.



- Lightweight DBM: `ldbm`.
- Entre outros.

```
# global configuration directives
<global config directives>

# backend definition
backend <typeA>

<backend-specific directives>

# first database definition & config directives
database <typeA>

<database-specific directives>
```



## Diretivas database

`database <type>` (marca o início da definição de uma nova instância de base de dados).

`<type>` (pode ser de qualquer tipo, como na diretiva `Backend`).

`readonly {on | off}` (coloca a base de dados no modo *read-only*).

```
replica host =<hostname>[:<port>]
[bindmethod={simple | kerberos | sasl}]
[“binddn=<DN>”]
```

(Especifica uma replicação para essa base de dados).

`relogfile <filename>` (especifica o nome do arquivo de log de replicações para o SLAPD que fará o log de mudanças).

Há outros tipos de diretivas:

## Diretivas database BDB

`directory <directory>` (indica o diretório onde residem os arquivos BDB contendo a base de dados e índices associados).

Por default: `<directory>` é `/usr/local/var/openldap-data`



Observe que essas diretivas se aplicam no caso em que o database considerado é BDB.

1. `# example config file – global configuration section.`
2. `include /usr/local/etc/schema/core.schema`
3. `referral ldap://root.openldap.org`
4. `access to * by * read`



## Exemplo de arquivo de configuração

O arquivo a seguir é um exemplo de `slapd.conf` que define duas bases de dados para manipular diferentes partes da árvore X.500, ambas instâncias da base de dados BDB.

```
# arquivo slapd.conf

1. # example config file - global configuration section
2. include /usr/local/etc/schema/core.schema
3. referral ldap://root.openldap.org
4. access to * by * read
```

- A linha 1 é um comentário.
- A linha 2 inclui outro arquivo de configuração que contém definições de esquemas.
- A linha 3 contém a diretiva *referral* para indicar consultas remotas para uma das bases de dados referenciadas no servidor na porta padrão 389 no host *root.openldap.org*.
- A linha 4 é um controle de acesso global.

```
6. database bdb
7. suffix "dc=example,dc=com"
8. directory /usr/local/var/openldap-data
9. rootdn "cn=Manager,dc=example,dc=com"
10. rootpw secret
20. index uid pres,eq
21. index cn,sn,uid pres,eq,approx,sub
22. index objectClass eq
```



```
# Continuação arquivo slapd.conf

5. # BDB definition for the example.com

6. database bdb
7. suffix "dc=example,dc=com"
8. directory /usr/local/var/openldap-data
9. rootdn "cn=Manager,dc=example,dc=com"
10. rootpw secret
11. # replication directives
12. replogfile /usr/local/var/openldap/slapd.replog
13. replica host=slave1.example.com:389
14. binddn="cn=Replicator,dc=example,dc=com"
15. bindmethod=simple credentials=secret
16. replica host=slave2.example.com
17. binddn="cn=Replicator,dc=example,dc=com"
18. bindmethod=simple credentials=secret
19. # indexed attribute definitions
```



```

20. index uid pres,eq
21. index cn,sn,uid pres,eq,approx,sub
22. index objectClass eq
23. # database access control definitions
24. access to attr=userPassword
25. by self write
26. by anonymous auth
27. by dn="cn=Admin,dc=example,dc=com" write
28. by * none
29. access to *
30. by self write
31. by dn="cn=Admin,dc=example,dc=com" write
32. by * read

```

```

34. database bdb
35. suffix "dc=example,dc=net"
36. directory /usr/local/var/openldap-data-net
37. rootdn "cn=Manager,dc=example,dc=com"
38. index objectClass eq
39. access to * by users read

```



```

# Continuação arquivo slapd.conf
33. # BDB definition for example.net
34. database bdb
35. suffix "dc=example,dc=net"
36. directory /usr/local/var/openldap-data-net
37. rootdn "cn=Manager,dc=example,dc=com"
38. index objectClass eq
39. access to * by users read

```

Essa forma de configuração está sendo descontinuada nas novas versões do LDAP. Em seu lugar está sendo criado o arquivo *slapd-config* (formato *cn=config*). Nesse novo formato serão usados arquivos tipo LDIF, que não devem ser editados ou ter seu conteúdo alterado diretamente: sempre devemos usar as ferramentas do LDAP: *ldapadd*, *ldapdelete* ou *ldapmodify*.

## Convertendo slapd.conf para o formato cn=config

- A conversão pode ser feita utilizando o aplicativo slaptest.
- `slaptest -f /usr/local/etc/openldap/slapd.conf -F /usr/local/etc/openldap/slapd`
- Teste a conversão com o rootdn e rootpw conforme a seguir:
  - ▣ `ldapsearch -x -D cn=config -w VerySecret -b cn=config`



## Executando o servidor LDAP

- SLAPD é projetado para executar como servidor stand alone.
- Vantagens como caching e gerenciamento de concorrência.
- Opções de linhas de comando:

```
-f <filename>
-n <service-name>
-u user -g group
-d <level> | ?
```

- Inicialização do servidor LDAP:

```
# /usr/local/etc/libexec/slapd [<option>]
```

- Encerramento do servidor LDAP:

```
# kill -INT `cat /usr/local/var/slapd.pid`
```



O daemon SLAPD é projetado para executar como um servidor stand alone, permitindo vantagens como caching e gerenciamento de concorrência, entre outras.

Algumas das opções que podem ser passadas para SLAPD:

- **f <filename>**: especifica um arquivo de configuração alternativo para SLAPD; o default é normalmente `/etc/openldap/slapd.conf`;
- **n <service-name>**: especifica o nome do serviço usado para login e outros propósitos. O nome pode ser os nomes, ou UID e GID, respectivamente;
- **d <level>**: indica o nível de debug para visualização dos logs associados à execução do SLAPD; do serviço default é `slapd`;
- **u user -g group**: indica usuário e grupo, respectivamente, para executar SLAPD; `user` e `group`.

## Inicializando o servidor LDAP

Para executar o servidor, utilize o comando:

```
# /etc/libexec/slapd [<option>] (option é uma das opções de linha de comando vistas anteriormente)
```

## Encerrando o servidor LDAP

Para encerrar o serviço de maneira segura, use o comando:

```
# kill -INT `cat /usr/local/var/slapd.pid`
```

Nesse momento, vale ressaltar que o SLAPD escreve seu PID no arquivo chamado `slapd.pid` no diretório configurado em `slapd.conf`. Ele também escreve seus argumentos no arquivo chamado `slapd.args`, no diretório configurado em `slapd.conf`, por exemplo, em `/usr/local/var/slapd.args`.



## Criação e manutenção de base de dados

On-line:

- Geralmente utilizado para adicionar poucas entradas.

Off-line:

- Geralmente usado para adicionar grandes bases de dados.
- Usa utilitários especiais providos pelo SLAPD.
- Faz leitura de *slapd.conf* e do arquivo LDIF.
- Produz os arquivos de índice da base de dados de maneira direta.

Há duas maneiras de criar uma base de dados. Pode-se criar uma base de dados on-line usando LDAP. Com esse método, é inicializado SLAPD e adicionadas entradas usando o cliente LDAP escolhido. A segunda maneira é fazer essa criação off-line, usando utilitários especiais providos pelo SLAPD. Esse método é melhor quando estamos trabalhando com bases de grande porte ou especificamente se desejamos garantir que a base de dados não seja acessada enquanto está sendo criada.

### Criando uma base de dados on-line

O pacote OpenLDAP vem com um utilitário chamado *ldapadd*, usado para adicionar entradas enquanto o servidor LDAP está executando. Nesse caso, devemos configurar algumas opções em *slapd.conf* antes de iniciar SLAPD.

- `suffix <dn>` (indica as entradas que serão mantidas pela base de dados.)
- `directory /usr/local/tudelft` (indica onde os arquivos índice devem ser criados.)
- `rootdn <dn>` e `rootpw <passwd>` (indicam um DN e senha que podem ser usados para autenticar a entrada de super-usuário na base de dados. Se o SASL está sendo usado como um mecanismo para autenticação em vez de LDAP, a linha `rootpw` pode ser descartada.)
- `index {<attrlist> | default} [press,eq,aprox,sub,none]` (indica as definições de índice desejadas.)
- Entre outros.

On-line:

- `ldapadd -f entrada.ldif -x -D "\\cn=Admin,dc=example,dc=com" -w secret`
- Deste modo:
  - `rootdn`
    - `cn=Admin, dc=example, dc=com`
  - `rootpw` como “secret”

Após especificar tais opções, use o comando:

```
# ldapadd -f /tmp/newentry -x -D "cn=Admin, dc=example, dc=com" -w secret
```

Essa linha assume `rootdn` como “`cn=Admin, dc=example, dc=com`” e `rootpw` como `secret` (pode ser uma senha encriptada SHA-1 em *slapd.conf*). Caso não deseje especificar a senha na linha de comando, use a opção `-W` em vez de `-w`.

## Criando base de dados off-line

Para criar off-line, são necessárias algumas ferramentas que fazem a leitura do arquivo de configuração *slapd.conf* e uma entrada no arquivo LDIF contendo uma representação das entradas a adicionar. Elas produzem os arquivos de índice da base de dados, de maneira direta. Opções como *suffix*, *directory* e *index* devem ser especificadas.

Após isso, use o comando:

```
# slapadd -l <inputfile> -f <slapdconfigfile> [-d <debuglevel>] [-n <integer>] | -b <suffix>]
```

*-l <inputfile>* (especifica o arquivo LDIF contendo as entradas a adicionar no formato texto).

*-f <slapdconfigfile>* (indica o arquivo de configuração SLAPD que cria os índices).

*-d <debuglevel>* (indica o nível de debug).

*-n <databasenum>* ou *-b <suffix>* (indicam a base de dados a modificar).

Off-line:

- **slapadd**: entrada da base de dados.
- **slapdindex**: geração dos arquivos de índice.
- **slapcat**: copia o conteúdo da base de dados em um arquivo no formato LDIF.

Às vezes, é necessário gerar índices novamente através do comando:

```
# slapindex -f <slapdconfigfile> [-d <debuglevel>] [-n <databasenum>] | -b <suffix>]
```

O programa *slapcat* é usado para fazer dump da base de dados para um arquivo LDIF e sua sintaxe é:

```
# slapcat -l <filename> [-d <debuglevel>] [-n <databasenum>]
```

É importante ressaltar que LDIF é usado para representar entradas LDAP em formato de texto simples. A forma básica de uma entrada é:

```
# comment  
dn: <distinguished name>  
<attrdesc>:<attrvalue>  
<attrdesc>:<attrvalue>
```

Alguns utilitários LDAP:

- **ldapsearch**: pesquisa entradas nas bases de dados LDAP.
- **ldapdelete**: remove entradas das bases de dados LDAP.
- **ldapmodify**: modifica entradas nas bases de dados LDAP.

*ldapsearch* é uma interface Shell para a chamada de sistema *ldap\_search*, usada para pesquisar entradas nas bases de dados LDAP. Exemplo:

```
# ldapsearch -u -b "o=rnp,c=br" "cn=Pedro Silva" sn mail
```

Ldapdelete é uma interface shell para a chamada ldap\_delete, usada para remover entradas das bases de dados LDAP. Exemplo:

```
# ldapsearch "cn=Pedro Silva,o=rnp,c=br"
```

Ldapmodify é uma interface shell para as chamadas ldap\_modify e ldap\_add, usadas para modificar entradas nas bases de dados LDAP. Exemplo:

```
# ldapmodify -b -r -f /tmp/entrymods
```

## Informações e características adicionais

Ferramentas de migração LDAP:

- Coleção de scripts Perl usada para converter arquivos de configuração para o formato LDIF.
- Úteis no uso do servidor LDAP para autenticar usuários, converter NIS e arquivos de senha para LDIF.

A autenticação usando LDAP é suportada na operação de bind. Tipos de autenticação:

- Anônima.
- Simples.
- Autenticação SASL.

Coleção de scripts em Perl usados para converter arquivos de configuração para o formato LDIF, muito úteis para autenticar usuários via servidor LDAP. Podem ser usados também para converter NIS e arquivos de senhas para o formato LDIF, fazendo com que esses arquivos sejam compatíveis com o servidor LDAP.

Para acessar serviços LDAP, o cliente LDAP primeiro deve autenticar-se para o serviço, ou seja, deve informar ao servidor LDAP que está acessando os dados de modo que o servidor pode decidir o que o cliente poderá ver e fazer.

No LDAP, a autenticação é suportada na operação de bind. O LDAP suporta três tipos de autenticações: anônima, simples e SASL. Um cliente que envia um pedido de autenticação sem fazer um bind é tratado como um cliente anônimo.

A autenticação simples consiste em enviar ao servidor LDAP o DN completamente qualificado do cliente e sua senha (não criptografada).

A autenticação Simple Authentication and Security Layer (SASL) especifica um protocolo cujos dados são transferidos entre o cliente e o servidor, utilizando canais encriptados (como SSL) suportados pelo servidor LDAP, que diminuem a exposição da senha. Com SASL, o LDAP pode suportar qualquer tipo de autenticação combinada entre o cliente e o servidor LDAP.

Ferramentas LDAP gráficas:

- KDirAdm.
- Directory Administrator.
- GQ.
- LDAP Browser.
- **KDirAdm**: ferramenta de gerência de diretórios LDAP escrita para KDEv2 e mais recentes;
- **Directory Administrator**: aplicação GNOME para gerenciamento de usuários e grupos;
- **GQ**: cliente LDAP gráfico escrito para GNOME;
- **LDAP Browser/Editor**: ferramenta para administração e visualização.



# Roteiro de Atividades 4

## Atividade 4.1 – Instalação do servidor OpenLDAP

Será realizada a instalação do servidor LDAP a partir do repositório Debian.

### 1. Instale os seguintes aplicativos:

```
# apt-get install slapd ldap-utils db4.8-util acl attr migrationtools  
libpam-ldap libnss-ldap nscd gzip
```

Serão feitas as seguintes perguntas:

- Senha do admin: rnpesr
- Confirme a senha: rnpesr
- Identificador de Recurso Uniforme do Servidor LDAP; no servidor, informe: ldap://127.0.0.1.
- O nome distinto da base de procura: dc=empresa,dc=com,dc=br
  - ▣ Versão LDAP: 3
  - ▣ Conta LDAP para o root: cn=admin,dc=empresa,dc=com,dc=br
  - ▣ Password da conta root do LDAP: rnpesr
  - ▣ Tornar a conta root local o administrador da base de dados? Sim.
  - ▣ A base de dados LDAP requer autenticação? Não.
  - ▣ Conta LDAP para o root: cn=admin,dc=empresa,dc=com,dc=br
  - ▣ Password da conta root do LDAP: rnpesr

### 2. Reconfiguração do pacote SLAPD:

```
#dpkg-reconfigure slapd
```

Serão feitas as seguintes perguntas:

- Omitir a configuração do Servidor OpenLdap? *Não.*
- Informe o nome de domínio DNS: *empresa.com.br*
- Informe o nome de sua organização: *Empresa.*
- Senha do admin: rnpesr
- Confirme a senha: rnpesr
- Backend da base de dados: BDB
- Você deseja que sua base de dados seja removida? *Não.*
- Mover a base de dados antiga? *Sim.*
- Permitir protocolo LDAPv2? *Não.*

### 3. Inicie o servidor LDAP:

```
# /etc/init.d/slapd start
```

### 4. Depois de iniciado o servidor LDAP, podemos testá-lo com a seguinte consulta:

```
# ldapsearch -x -b '' -s base "(ObjectClass=*)"
```





5. Se aparecer no console o resultado a seguir, está tudo certo:

```
# extended LDIF
# LDAPv3
# base<> with scope base
# filter: (ObjectClass=*)
# requesting: ALL
#
dn:
objectClass: top
objectClass: OpenLDAProotDSE
# search result
search: 2
result: 0 Success
# numResponses: 2
# numEntries: 1
```

## Atividade 4.2 – Usando o Migration Tools

Migration Tools é um conjunto de scripts que permite importar as contas locais do Sistema Operacional Linux para o LDAP.

1. Edite o arquivo `/usr/share/migrationtools/migrate_common.ph` e substitua “padl.com” pelo seu domínio “empresa.com.br”, e “dc=padl,dc=com” por “dc=empresa,dc=com,dc=br”.

```
# Default DNS domain
$DEFAULT_MAIL_DOMAIN = “empresa.com.br”;
# Default base
$DEFAULT_BASE = “dc=empresa,dc=com,dc=br”;
```

2. Gere as bases de dados para exportação:

```
# cd /usr/share/migrationtools
# ./migrate_base.pl > /tmp/base.ldif
# ./migrate_passwd.pl /etc/passwd /tmp/passwd.ldif
# ./migrate_group.pl /etc/group /tmp/group.ldif
```

3. Edite o arquivo `/tmp/base.ldif` e remova os registros:

```
dn: dc=com,dc=br
dc: com
objectClass: top
```

```
objectClass: domain
dn: dc=empresa,dc=com,dc=br
dc: empresa
objectClass: top
objectClass: domain
```

#### 4. Adicione na base LDAP os registros:

```
# ldapadd -x -W -D 'cn=admin,dc=empresa,dc=com,dc=br' < /tmp/base.
ldif

# ldapadd -x -W -D 'cn=admin,dc=empresa,dc=com,dc=br' < /tmp/
passwd.ldif

# ldapadd -x -W -D 'cn=admin,dc=empresa,dc=com,dc=br' < /tmp/group.
ldif

# /etc/init.d/slaped restart
```

### Atividade 4.3 – Configuração do cliente Linux para uso do LDAP

Para que as máquinas possam se autenticar na base de dados do LDAP, será necessário configurar o PAM e o NSS para consultarem o servidor do LDAP. Vamos configurar o servidor e o cliente Linux para se autenticarem na base do LDAP, que está rodando na máquina virtual servidor (192.168.1.1).

#### Instalação dos pacotes

No servidor Linux os pacotes já foram instalados na Atividade 1.

No cliente Linux:

```
# apt-get install libnss-ldap libpam-ldap nscd
```

Responda as perguntas informando o endereço do servidor do LDAP:

- Endereço para consulta de usuários no LDAP? *ldap://192.168.1.1/*
- Nome da base de pesquisa? *dc=empresa,dc=com,dc=br*
- Versão do protocolo LDAP a ser utilizada? *3*
- Conta LDAP para o root? *cn=admin,dc=empresa,dc=com,dc=br*
- Senha da conta root do LDAP? *Rnpesr*
- Privilégios especiais LDAP para o root? *Sim*
- A base de dados LDAP requer autenticação? *Não*

#### Configurando a autenticação pelo LDAP

Devemos criar ou editar os arquivos */etc/libnss-ldap.secret* e */etc/pam\_ldap.secret* contendo a Password Root DN como texto puro, ou seja, apenas uma linha com “senha”, sem as aspas. Isso facilitará o administrador do LDAP via linha de comando, pois não será necessário informar a senha toda vez que se fizer alguma alteração na base LDAP. Mas tome cuidado com esse arquivo, pois ninguém que estiver logado no servidor poderá lê-lo.

1. Verifique o conteúdo do arquivo */etc/libnss-ldap.secret*. Caso não exista, insira a senha do servidor LDAP:

```
rnpesr
```

2. Mudando a permissão do arquivo */etc/libnss-ldap.secret*:

```
# chmod 600 /etc/libnss-ldap.secret
```

3. Verifique o conteúdo do arquivo */etc/pam\_ldap.secret*. Caso não exista, insira a senha do servidor LDAP:

```
rnpesr
```

4. Mudando a permissão do arquivo */etc/pam\_ldap.secret*:

```
# chmod 600 /etc/pam_ldap.secret
```

5. Verifique a configuração do arquivo */etc/libnss-ldap.conf*

```
base dc=empresa,dc=com,dc=br
uri ldap://192.168.1.1
ldap_version 3
rootbinddn cn=admin,dc=empresa,dc=com,dc=br
```

6. Verifique a configuração do arquivo */etc/pam\_ldap.conf*

```
base dc=empresa,dc=com,dc=br
uri ldap:// 192.168.1.1
ldap_version 3
rootbinddn cn=admin,dc=empresa,dc=com,dc=br
```

7. Edite o arquivo */etc/ldap/ldap.conf*

```
BASEdc=empresa,dc=com,dc=br
URI ldap://192.168.1.1
```

8. Precisamos agora configurar o Name Service Switch (NSS) para usar LDAP através do arquivo */etc/nsswitch.conf*:

```
passwd:    compat ldap
group:     compat ldap
shadow:    compat ldap
hosts:     files dns
networks:  files
protocols: db files
services:  db files
ethers:    db files
rpc:       db files
netgroup:  ldap
```

9. Reinicie o serviço NSCD:

```
# /etc/init.d/nscd restart
```

10. Para usar o LDAP, as configurações do Plugabe Authentication Module (PAM) devem ser alteradas. O diretório de configuração é */etc/pam.d/*. Edite o arquivo */etc/pam.d/common-password* e remova o atributo *use\_authok*:

```
password [success=1 user_unknown=ignore default=die] pam_ldap.so  
try_first_pass
```

11. Edite o arquivo */etc/pam.d/common-session* e inclua a linha a seguir no final do arquivo:

```
session optional pam_mkhomedir.so skel=/etc/skel/ umask=077
```

12. O teste pode ser feito com a ferramenta *getent*, da seguinte forma:

```
# getent passwd  
# getent shadow  
# getent group
```

Se você visualizar o conteúdo semelhante aos arquivos locais (*/etc/passwd*, */etc/shadow*, */etc/group*) é porque não há nenhuma informação extra no LDAP.

## Atividade 4.4 – Criação e teste de base de dados para ser adicionada ao servidor OpenLDAP

---

1. Crie o arquivo */root/aluno2.ldif*:

```
dn: uid=aluno2,ou=People,dc=empresa,dc=com,dc=br  
uid: aluno2  
cn: aluno2  
objectClass: account  
objectClass: posixAccount  
objectClass: top  
objectClass: shadowAccount  
userPassword: {crypt}$1$9Dt/Yo20$hou4BMnIf9S8Cn70RcAyg0  
shadowLastChange: 13865  
shadowMax: 99999  
shadowWarning: 7  
loginShell: /bin/bash  
uidNumber: 1005  
gidNumber: 1010  
homeDirectory: /home/aluno2  
gecos: aluno2,,,
```



2. Crie o arquivo */root/grupo.ldif*:

```
dn: cn=rnp,ou=Group,dc=empresa,dc=com,dc=br
objectClass: posixGroup
objectClass: top
cn: rnp
userPassword: {crypt}x
gidNumber: 1010
```

3. Adicione o arquivo */root/grupo.ldif* à base:

```
# ldapadd -x -D "cn=admin,dc=empresa,dc=com,dc=br" -w rnpesr -f /
root/grupo.ldif
```

4. Adicione o arquivo */root/aluno2.ldif* à base:

```
# ldapadd -x -D "cn=admin,dc=empresa,dc=com,dc=br" -w rnpesr -f /
root/aluno2.ldif
```

5. Troque a senha do usuário "aluno2":

```
# passwd aluno2
```

## Atividade 4.5 – Testando o funcionamento do LDAP

---

1. Na máquina cliente, abra um novo console ou terminal e tente logar como usuário do LDAP criado a partir do arquivo:

```
login: aluno2
```

2. Verifique no arquivo */etc/passwd* do servidor se o usuário do LDAP existe:

```
# cat /etc/passwd
```

3. Você não deve ter encontrado o usuário "aluno2". Para encontrá-lo, execute:

```
# getent passwd
```

4. Com os grupos ocorre a mesma coisa; por meio do arquivo */etc/group* não é possível visualizar o grupo RNP criado no LDAP. Para vê-lo, use:

```
# getent group
```

# 5

## DHCP, Telnet, FTP e SSH

### objetivos

Aprender teoria e prática do serviço DHCP, conhecer formas de transferência de arquivos entre computadores, entender o funcionamento do File Transfer Protocol (FTP), analisar formas seguras de transferência de arquivos, envolvendo encriptação e autenticação, inclusive na administração remota de outros servidores e backup.

DHCP, Telnet, FTP, serviço SSH.

### conceitos

### Introdução

- Dynamic Host Configuration Protocol (DHCP).
  - ▣ Instalação do servidor DHCP.
  - ▣ Configurando e iniciando o servidor DHCP.
- Configurando e iniciando o servidor Telnet.
- File Transfer Protocol (FTP).
- Secure Shell (SSH).
  - ▣ Instalando e configurando o servidor SSH.
  - ▣ Geração e utilização de chaves de autenticação.
  - ▣ Transferência de arquivos segura utilizando SCP e SFTP.



O estudo deste capítulo está dividido em três partes: DHCP, Telnet e SSH. Na primeira parte são abordados aspectos teóricos e práticos do serviço DHCP, sendo examinadas algumas possibilidades de uso dessa solução. Na segunda parte veremos a teoria relacionada ao Telnet, um importante serviço utilizado em muitas redes, com uma abordagem das configurações práticas desse serviço.

Na terceira parte serão estudadas as formas de transferência de arquivos entre computadores, como o serviço FTP, associadas a grupos de usuários e pastas. Conheceremos formas seguras de transferência de arquivos, que envolvem encriptação e autenticação, inclusive na administração remota de outros servidores e backup.



## DHCP

- Protocolo de rede cuja função é atribuir informações TCP/IP para as máquinas clientes.
- As informações são concentradas em um servidor (DHCP), que passa para as máquinas informações como:
  - ▣ Endereço IP e máscara de rede.
  - ▣ Gateway.
  - ▣ Servidor DNS.
  - ▣ Domínio.



O protocolo Dynamic Host Configuration Protocol (DHCP) provê, através de um servidor DHCP, informações para estações IP em uma rede. Consiste em dois componentes: um protocolo para distribuir parâmetros de configuração específicos a partir de um servidor DHCP; e um mecanismo para alocação de endereços de rede para as estações. É baseado na arquitetura cliente e servidor, onde o servidor DHCP distribui endereços IP e parâmetros de configuração de rede para as estações, tais como: máscara de sub-rede; endereço do roteador de saída (gateway); informações de domínio e outras que serão mencionadas adiante.

Existem alguns protocolos que proveem recursos utilizados pelo protocolo DHCP, tais como: BOOTP; Reverse Address Resolution Protocol (RARP), através do Dynamic RARP (DRARP); Trivial File Transfer Protocol (TFTP) e Internet Control Message Protocol (ICMP).

- **BOOTP**: mecanismo de transporte de informações de configuração. Permite que estações diskless obtenham um endereço IP, além do endereço IP do servidor BOOTP, e um arquivo a ser carregado em memória para fins de boot. Pode ser utilizado ainda como relay para servidores DHCP, eliminando a necessidade de um servidor DHCP em cada rede física.
- **RARP**: protocolo que permite a obtenção de um endereço IP a partir de um endereço MAC. Uma estação envia uma mensagem em broadcast na rede com a finalidade de que o servidor DHCP lhe forneça um endereço IP.
- **TFTP**: protocolo utilizado para transferência de arquivos, no qual cada pacote é confirmado (ACK) individualmente.
- **ICMP**: protocolo utilizado para permitir que estações localizem roteadores por meio de mensagens ICMP Redirect.

Motivação:

- Ferramenta que facilita a administração de redes.

Funcionalidades:

- Controla a forma de atribuição de endereços IP às máquinas cliente da rede.
- Estabelecimento de faixas de endereços utilizados.
- Informações para as máquinas clientes referentes aos endereços do roteador de saída (gateway), servidor DNS e domínio.



O serviço DHCP foi projetado para suportar as RFCs associadas aos requisitos para funcionamento das estações (host requirements). Após a obtenção dos parâmetros por meio do protocolo DHCP, as estações devem ser capazes de trocar pacotes com qualquer outra na internet.



O DHCP suporta três mecanismos para alocação de endereços IP:

- Alocação automática.
- Alocação dinâmica.
- Alocação manual.

O DHCP suporta três mecanismos para alocação de endereços IP:

- **Alocação automática:** um endereço IP é atribuído permanentemente para uma estação;
- **Alocação dinâmica:** um endereço IP é atribuído para uma estação por um período determinado de tempo;
- **Alocação manual:** um endereço IP é atribuído para uma estação pelo administrador da rede, o que é feito por meio do arquivo de configuração do DHCP.

Pode-se utilizar um ou mais desses mecanismos ao mesmo tempo em uma determinada rede. Uma estação cliente deve ser capaz de descobrir os parâmetros necessários ao seu funcionamento, inserindo-os automaticamente em seu sistema sem intervenção manual. Do mesmo modo, o servidor deve funcionar de maneira automática sem a necessidade de intervenção manual do administrador para o funcionamento de cada estação cliente, exceto em casos especiais onde seja necessário fixar determinado endereço a um cliente específico. Deve ainda suportar estações que utilizem o protocolo BOOTP (RFC 2132), IPv6 (RFC 4361), SIP (RFC 3319) e IEEE 1394 (RFC 2855).

### Formato de uma mensagem DHCP

|                    |           |           |          |
|--------------------|-----------|-----------|----------|
| OP (1)             | HTYPE (1) | HLEN (1)  | HOPS (1) |
| XID (4)            |           |           |          |
| SECS (2)           |           | FLAGS (2) |          |
| CIADDR (4)         |           |           |          |
| YIADDR (4)         |           |           |          |
| SIADDR (4)         |           |           |          |
| GIADDR (4)         |           |           |          |
| CHADDR (16)        |           |           |          |
| SNAME (64)         |           |           |          |
| FILE (128)         |           |           |          |
| OPTIONS (variável) |           |           |          |

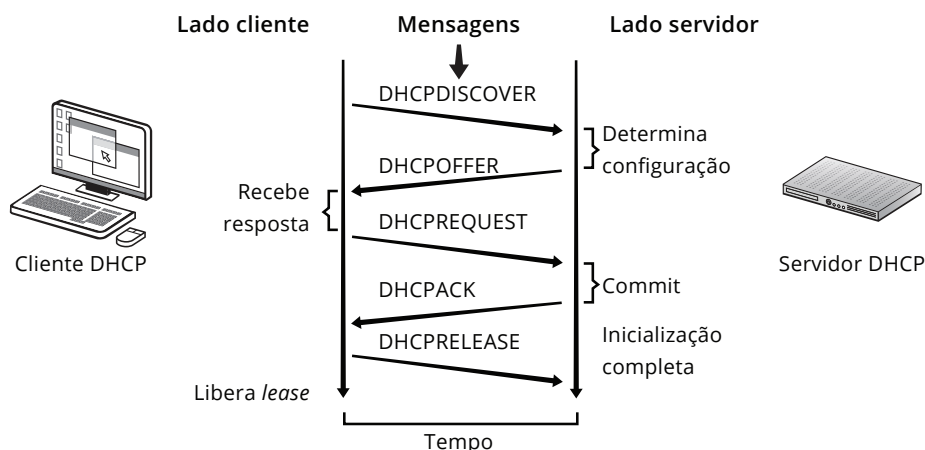
**Figura 5.1**  
Formato de  
mensagens DHCP  
indicando campos e  
tamanho em bytes.



Campos em uma mensagem DHCP:

- **OP**: numa mensagem DHCP, uma solicitação e uma resposta possuem os mesmos campos. O que as diferenciam é o conteúdo desse campo. A informação 1 indica uma solicitação (BOOTREQUEST), a informação 2 indica uma resposta (BOOTREPLY);
- **HTYPE (Hardware Address Type)**: informa o padrão de rede utilizado pelo adaptador de rede;
- **HLEN (Hardware Address Length)**: valor 6 para endereço MAC;
- **HOPS**: quantidade de roteadores pelos quais a mensagem deverá passar;
- **XID (Transaction ID)**: número aleatório escolhido pelo cliente, utilizado em solicitações e respostas associadas;
- **SECS**: segundos desde que o cliente iniciou processo de aquisição ou renovação de endereço;
- **FLAGS**: primeiro bit 1 para broadcast; demais reservados para uso futuro;
- **CIADDR (Client IP Address)**: preenchido somente no cliente que está em processo de renovação;
- **YIADDR (Your Client IP Address)**: endereço IP cliente;
- **SIADDR**: endereço IP do próximo servidor a ser utilizado no processo de boot; retornado pelo servidor;
- **GIADDR**: endereço IP do roteador da rede local, utilizado quando o boot é realizado por meio deste;
- **CHADDR (Client Hardware Address)**: endereço de hardware do cliente;
- **NAME**: Server Host Name (opcional);
- **FILE**: nome do arquivo de boot; nulo em mensagem DHCPDISCOVER; caminho completo de arquivo em DHCPOFFER;
- **OPTIONS**: parâmetros opcionais. Esse campo é utilizado para informar que tipo de resposta ou solicitação DHCP (DHCPDISCOVER, DHCPOFFER etc.) está sendo enviada para o cliente ou para o servidor.

## Funcionamento do protocolo DHCP



**Figura 5.2**  
Funcionamento do protocolo DHCP.

Durante a obtenção dos parâmetros de configuração necessários, algumas mensagens são trocadas entre o servidor e o cliente DHCP. A seguir é descrita a sequência desses comandos com seus respectivos significados:

- Cliente envia mensagem broadcast DHCPDISCOVER. Inclui o endereço físico, podendo incluir sugestão de endereço IP e duração do lease.
- O servidor pode responder com uma mensagem DHCPOFFER, que inclui um endereço IP disponível no campo YIADDR e outros parâmetros em OPTIONS. O servidor verifica a disponibilidade do endereço IP antes de disponibilizá-lo.
- O cliente envia uma mensagem DHCPREQUEST que inclui o identificador do servidor DHCP. Isso é necessário para o caso de o cliente receber respostas de mais de um servidor DHCP.
- O servidor, após receber a mensagem, salva as configurações e responde com uma mensagem DHCPACK contendo as configurações ofertadas anteriormente.
- O cliente efetua uma verificação utilizando o protocolo ARP com o endereço fornecido. Caso perceba que o endereço já está em uso, envia uma mensagem DHCPDECLINE. Caso o cliente receba uma mensagem DHCPNACK, o processo é reiniciado.
- O cliente pode ainda liberar o endereço informando seu CHADDR.

Caso o cliente já saiba o endereço IP, desejando apenas renová-lo, enviará diretamente um DHCPREQUEST.

Verificações iniciais:

- Verificar se o daemon dhcpd está instalado.
- Verificar a existência do arquivo *dhcpd.leases*.
- Escolher o esquema de interação com o serviço Domain Name Service (DNS).



É necessário efetuar algumas verificações antes da edição do arquivo de configuração e ativação do servidor DHCP. Entre elas: verificar se o daemon está instalado e verificar a sua localização, assim como a existência do arquivo com a base de dados e a forma de interação com o serviço DNS.

#### 1. Verificar se o daemon está instalado:

Verificar localização do daemon dhcpd.

```
# whereis dhcpd
```

#### 2. Criação do arquivo *dhcpd.leases*:

Verificar existência do arquivo *dhcpd.leases*, que deverá existir quando da execução do servidor DHCP.

```
# find / -name dhcpd.leases
```

Deve retornar algo do tipo: */var/lib/dhcp/dhcpd.leases*.

Caso o arquivo *dhcpd.leases* não exista, deverá ser criado.

```
# touch /var/lib/dhcp/dhcpd.leases
```

#### 3. Escolha do esquema de interação com o DNS:

Existem dois métodos de interação entre o DHCP e o DNS:

- **ddns-update-style interim:** permite a utilização de mecanismo *failover*. Esse mecanismo possibilita que dois ou mais servidores DHCP dividam o conjunto de endereços IP disponíveis.

- **ddns-update-style ad-hoc:** opção desatualizada, que não deve ser utilizada.

Deve-se escolher o método a ser utilizado pelo servidor. A opção deve ser incluída no início do arquivo `/etc/dhcpd.conf`.

## Edição de arquivos de configuração

Inicialmente deve-se criar o arquivo de configuração `dhcpd.conf`, que consiste em:

- Informações de cabeçalho.
- Parâmetros.
- Declarações.
- Comentários.

As opções podem ser:

- Globais, posicionadas antes das `{ }`.
- Específicas para cada cliente.

O arquivo de configuração `dhcpd.conf` deverá ser editado, incluindo-se nele os valores passados como parâmetros de configuração às estações cliente. Possui uma estrutura com quatro componentes: cabeçalho, parâmetros, declarações e comentários.

- **Cabeçalho:** esquema de interação entre DHCP e DNS, conforme visto anteriormente;
- **Parâmetros:** como, quando e quais informações devem ser enviadas aos clientes. Os parâmetros iniciados pela palavra *option* não são obrigatórios e não interferem no funcionamento do servidor; no entanto, a ausência de alguns deles pode causar problemas para as estações clientes;
- **Declarações:** informações sobre a topologia da rede, incluindo sub-redes, endereços fixos destinados a clientes específicos e especificação de domínio;
- **Comentários:** linhas iniciadas com `#` são consideradas como comentários.



Quaisquer mudanças nos arquivos de configuração somente terão efeito após a reinicialização do servidor.

Exemplo de arquivo de configuração com alguns valores típicos:

```
# Arquivo dhcpd.conf
ddns-update-style interim;

subnet 192.168.0.0 netmask 255.255.255.0 {
    option routers 192.168.0.254;
    option subnet-mask 255.255.255.0;
    option domain-name "linux.lan";
    option domain-name-servers 192.168.0.253;
    option time-offset -18000;
    range 192.168.0.2 192.168.0.250;
}
```

O conteúdo do arquivo de configuração apresentado indica:

- ▣ Interação com DNS: `ddns-update-style interim`.
- ▣ Servidor responsável por distribuir informações na rede: `192.168.0.0/255.255.255.0`
- ▣ Parâmetros *option* utilizados:
  - ▣ **routers**: indica gateway.
  - ▣ **subnet mask**: máscara de sub-rede.
  - ▣ **domain name**: nome do domínio.
  - ▣ **domain name servers**: endereço do servidor DNS.
  - ▣ **time offset**: tempo de lease, representado em segundos; o valor `0xffffffff` é utilizado para representar o infinito.

## Base de dados

Podem ainda ser utilizadas:

- ▣ Redes compartilhadas.
- ▣ Grupos.
- ▣ Endereços IP estáticos, utilizando DHCP.



Outro exemplo de arquivo de configuração, neste caso para duas sub-redes:

```
# Arquivo dhcpd.conf
# Redes compartilhadas
ddns-update-style interim;
shared-network name {
    option routers 192.168.1.254;
    option domain-name "linux.lan";
    option domain-name-servers 192.168.0.253;
    more parameters for rede2 shared-network
subnet 192.168.0.0 netmask 255.255.255.240 {
    parameters for subnet
    range 192.168.0.2 192.168.0.14;
}
subnet 192.168.0.16 netmask 255.255.255.240 {
    parameters for subnet
    range 192.168.0.17 192.168.0.30;
}
}
```



Trecho de arquivo de configuração:



```
subnet 192.168.0.0 netmask 255.255.255.240 {
    range 192.168.0.2 192.168.0.14;
}

subnet 192.168.0.0 netmask 255.255.255.240 {
    range 192.168.0.17 192.168.0.30;
}

host onca {
    option host-name "onca.linux.lan";
    hardware ethernet 00:D0:B7:89:7E:5E;
    fixed address 192.168.0.2;
}
```

A seguir é apresentado outro exemplo de arquivo de configuração, neste caso com a inclusão de grupo e de IP fixos para determinadas máquinas da rede:

```
# Arquivo dhcpd.conf

ddns-update-style interim;

group {
    option routers 192.168.1.254;
    option subnet mask 255.255.255.0;
    option domain-name "rede.com";
    option domain-name-servers 192.168.0.253;
    option time-offset -18000;

    host onca {
        option host-name "onca.linux.lan";
        hardware ethernet 00:D0:B7:89:7E:5E;
        fixed address 192.168.0.2;
    }

    host tigre {
        option host-name "tigre.linux.lan";
        hardware ethernet 00:D0:B7:89:01:D1;
        fixed address 192.168.0.3;
    }
}
```

## Iniciando e parando o servidor

- Red Hat/Fedora: `service dhcp <parâmetro>`
  - ▣ Parâmetro: `start`, `stop` ou `restart`.
- Debian: `/etc/init.d/dhcpd <parâmetro>`
  - ▣ Parâmetro: `start`, `stop` ou `restart`.

Observação: o arquivo `dhcpd.leases` deverá existir.

Quando o servidor dispõe de mais de uma interface de rede, pode-se especificar que este funcione somente para determinada interface. Essa funcionalidade é extremamente útil quando o servidor encontra-se conectado a uma rede interna a qual deve prover DHCP, e a uma rede externa a qual não se deseja acesso ao DHCP. Para isso, edite o arquivo `/etc/sysconfig/dhcpd` atribuindo a interface ao parâmetro `DHCPARGS`. Exemplo com a interface `eth0`:

```
# Arquivo /etc/sysconfig/dhcpd
# Command line options here
DHCPARGS=eth0
```

Podem ser passados parâmetros como:

- **p <porta>**: especifica a porta na qual o servidor DHCP atuará; padrão 67 e 68;
- **f <número da porta>**: executa o servidor em *foreground*;
- **d - modo debugging**: os logs são gerados para a tela, em vez de `/var/log/messages`;
- **cd <nome do arquivo>**: especifica a localização do arquivo de configuração `dhcpd.conf`. Originalmente em `/etc/dhcpd.conf`;
- **lf <nome do arquivo>**: especifica a localização do banco de dados do DHCP. Originalmente em `/var/lib/dhcp/dhcpd.leases`.

Configurando um cliente DHCP:

- A configuração pode ser efetuada utilizando-se linha de comando ou interface gráfica.
- Red Hat/Fedora.
  - ▣ Arquivo `/etc/sysconfig/network-scripts/ifcfg-eth0` se a interface a ser utilizada for `eth0`.
- Debian.
  - ▣ Arquivo `/etc/network/interfaces`

A configuração de um cliente DHCP pode ser efetuada em dois momentos: durante a instalação do Sistema Operacional; ou em qualquer momento se for realizada pelo usuário com privilégios de administrador da máquina.

Durante o processo de instalação do Sistema Operacional, surgirá uma mensagem solicitando que seja escolhida a opção de configuração da rede: automática (DHCP) ou manual. Escolhendo a opção automática, a máquina iniciará o processo para obtenção dos parâmetros de rede.

Após o processo de instalação, a configuração pode ser feita por meio de interface gráfica ou linha de comando. Para a configuração por linha de comando, o arquivo a ser editado é `/etc/sysconfig/network-scripts/ifcfg-eth0`, nas distribuições Red Hat/Fedora, ou no arquivo `/etc/network/interfaces`, no Debian.



## Exercício de fixação 1

### Instalação do servidor DHCP

Instale o servidor de DHCP na máquina virtual SERVIDOR, verifique se o serviço está funcionando e se o arquivo *dhcp.leases* foi criado. Se o arquivo *dhcp.leases* não foi gerado, será necessário criá-lo manualmente.

## Telnet

- O objetivo principal é a conexão a máquinas remotas de forma padronizada, tanto em rede local quanto pela internet.
- Provê comunicação bidirecional, orientada à conexão.
- Permite o acesso e a utilização de recursos computacionais da máquina remota.
- A desvantagem é que login, senha e dados trafegam em texto plano sem encriptação.
  - Recomenda-se utilizar SSH, deixando Telnet somente para onde não há cliente SSH.



O Telnet é um protocolo orientado à conexão, que se utiliza de Transmission Control Protocol (TCP), sendo utilizado para transmissão bidirecional de caracteres para uso geral, baseada em três conceitos principais:

- Concepção de um terminal virtual de rede;
- Princípio de opções negociadas;
- Visão simétrica de terminais e processos.

Quando uma conexão Telnet é estabelecida, cada extremidade é considerada a origem e a terminação de um terminal virtual de rede (Network Virtual Terminal – NVT). Um NVT é um dispositivo imaginário que provê uma representação de um terminal. Todos os hosts, tanto o cliente quanto o servidor, mapeiam as características de seus dispositivos locais de modo a parecerem um NVT, uns para os outros. Denomina-se servidor o lado que provê algum serviço, e cliente o lado que utiliza esse serviço e que normalmente inicia a comunicação.

Telnet usa uma codificação USA-ASCII de 7 bits em um campo de 8 bits. Apesar de o Telnet ser bidirecional e full-duplex, o NVT é visto como uma comunicação half-duplex com buffer. Os dados são acumulados até completar uma linha de dados, que é então transmitida.

A maior desvantagem do Telnet é que todo o tráfego na rede é em texto plano. Desse modo, todos os dados enviados, incluindo login e senha, podem ser interceptados para ferramentas de captura ativas na rede. Recomenda-se a utilização de Telnet somente quando não houver cliente SSH disponível. Outra possibilidade é a utilização de Telnet em redes internas seguras ou Virtual Private Network (VPN).

## Cliente Telnet

```
$ telnet <nome do servidor>
```

- open <nome do servidor>: inicia uma sessão.
- close: encerra sessão e retorna ao modo de comando.
- exit: encerra sessão e sai do Telnet.
- ?: exibe menu de ajuda.



Para que um host inicie uma conexão com um servidor, é necessário que o daemon telnetd esteja disponível e em execução. Uma sessão Telnet pode ser estabelecida entre máquinas independentemente dos Sistemas Operacionais, e pode ser utilizada para acessar servidores, roteadores, switches etc. Exemplo de conexão com Telnet:

```
$ telnet leao.linux.lan
Trying leao.linux.lan ...
Connected to leao.linux.lan (192.168.0.3)
Escape character is '^]'
Debian Sarge Kernel 2.6.15-3
login:
password:
Last login: Seg Out 16 09:32:54 from onca.linux.lan
$ netstat -nr
Kernel IP routing table
Destination      Gateway          Genmask         Flags
192.168.0.0      0.0.0.0         255.255.255.0   U
127.0.0.0        0.0.0.0         255.0.0.0       U
0.0.0.0          192.168.0.1     0.0.0.0         UG
$ exit
```

## Servidor Telnet

- Necessário instalar servidor Telnet.
  - ▣ No Red Hat: telnetd deve ser instalado.
  - ▣ No Fedora: telnetd deve ser instalado.
  - ▣ No Debian: telnetd deve ser instalado.
- Utiliza inetd (Debian) e xinetd (Red Hat/Fedora).
- Recomendável configurar para utilizar xinetd.

Versões antigas do Red Hat têm o telnetd instalado por padrão. Versões atuais Red Hat/Fedora devem ser instaladas manualmente.

```
Red Hat/Fedora
# rpm -ivh <pacote telnet>.rpm          (instala Telnet)
# chkconfig telnet on                  (ativa Telnet)
chkconfig telnet off                    (para desabilitar)
Debian
# apt-get install telnetd
```





No Debian o servidor Telnet utiliza inetd, em vez de xinetd. Para parar o servidor, deve-se comentar a linha do Telnet no arquivo */etc/inetd.conf*, reiniciando o servidor inetd em seguida com o comando:

```
# /etc/init.d/inetd restart
```

O xinetd possui maior flexibilidade que o inetd, por permitir a utilização de filtros de acesso baseados, por exemplo, em endereço IP de origem. Para trocar o inetd pelo xinetd, é necessário instalar o pacote xinetd e criar o arquivo de configuração */etc/xinetd.d/telnet*.

Formas de melhorar a segurança:

- Utilizar somente em redes locais seguras.
- Utilizar VPNs se dados trafegarem em redes públicas.
- Mudar a porta utilizada pelo servidor.
  - ▣ Melhora parcial.
  - ▣ Ferramentas portscan podem detectar a nova porta.
- Restringir o acesso por IP de origem.



Além da utilização do Telnet somente em redes locais seguras ou com VPN em redes públicas, pode-se ainda utilizar dois métodos adicionais para melhorar um pouco a segurança: troca da porta utilizada pelo servidor; restrição do acesso a alguns IPs.

```
# Arquivo /etc/xinetd.d/telnet
# default: on
# description: The telnet server serves telnet sessions
# unencrypted username/password pairs for authentication.
service telnet
{
    flags            = REUSE
    socket_type      = stream
    wait            = no
    user             = root
    server           = /usr/sbin/in.telnetd
    log_on_failure   += USERID
    disable          = no
    port            = 7777
    only_from        = 192.168.0.2 127.0.0.1 192.168.0.1
}
```

## File Transfer Protocol (FTP)

Transferência de arquivo confiável de/para servidores de arquivos.

- Upload: transferência de arquivos para um servidor.
- Download: transferência de arquivos de um servidor.

Tipos de arquivos transferidos:

- Arquivos texto: manipulados como sendo compostos por uma cadeia de caracteres ASCII ou EBCDIC.
- Arquivos binários: formados por uma sequência de octetos transferidos sem qualquer conversão.

O protocolo File Transfer Protocol (FTP) está entre os meios mais populares utilizados para transferência de arquivos pela internet. Grande número de servidores FTP está disponível tanto na internet quanto em redes locais, utilizados para distribuição de conteúdo, documentos etc. Muitos servidores web se utilizam do recurso FTP como meio para disponibilizar arquivos e programas. A facilidade de acesso aos dados e a confiabilidade provida pelo Transmission Control Protocol (TCP) são fatores que estimulam seu uso.

Todas as distribuições Linux vêm com um servidor FTP embutido, o qual pode estar ativado ou desativado. Existem ainda outros servidores FTP amplamente utilizados e que podem ser instalados, entre eles o VSFTPD e o PROFTPD.

Esse serviço pode ser configurado para permitir o acesso a todo e qualquer usuário, denominado anônimo, ou para restringir o acesso a determinados usuários. Em ambos os casos pode-se determinar quem acessa o quê, e os privilégios que possui nas pastas que acessa. Pode-se ainda efetuar ajustes para facilitar a passagem por meio de firewalls, para utilização de múltiplos servidores virtuais, e também para permitir a utilização por grupos de usuários.

Tipos de acesso:

- Por usuário normal.
  - ▣ Permite o acesso por parte de usuários cadastrados no servidor.
  - ▣ O usuário pode navegar no sistema de arquivos do servidor.
- Por usuário anônimo:
  - ▣ Permite o acesso por parte de qualquer usuário.
  - ▣ Restringe a área de acesso ao sistema de arquivos do servidor.

O administrador de um servidor FTP tem pleno controle sobre quem pode acessar o quê. Nesse caso, pode-se escolher entre permitir ou restringir o acesso de usuários anônimos. Pode-se ainda relacionar usuários cadastrados a determinadas pastas. Em todos os casos os privilégios de leitura, escrita, criação ou remoção de arquivos e pastas podem ser controlados.

De modo geral há dois tipos de usuários:

- **Usuário normal:** usuário cadastrado. Possui login e senha, e por meio deles acessa determinadas pastas e arquivos;
- **Usuário anônimo:** usuário que acessa o serviço sem ser autenticado. De modo geral possui somente permissão de leitura, podendo efetuar download de arquivos considerados públicos.

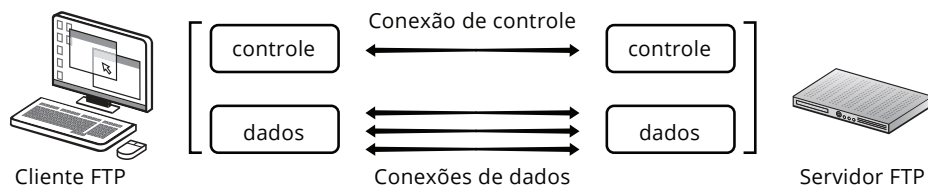


Nos dois casos podemos conceder privilégios de leitura (download) e escrita (upload) a determinadas pastas e arquivos. Entretanto, recomenda-se manter o usuário com acesso somente para leitura e na pasta pública. Caso não seja necessário um usuário anônimo, é desejável ainda bloquear o acesso a esse tipo de usuário.

## Funcionamento do FTP

Baseado no estabelecimento de conexões entre o cliente e o servidor:

- **Conexão de controle:** usada na transferência de arquivos.
- **Conexão de dados:** uma para cada arquivo transferido.

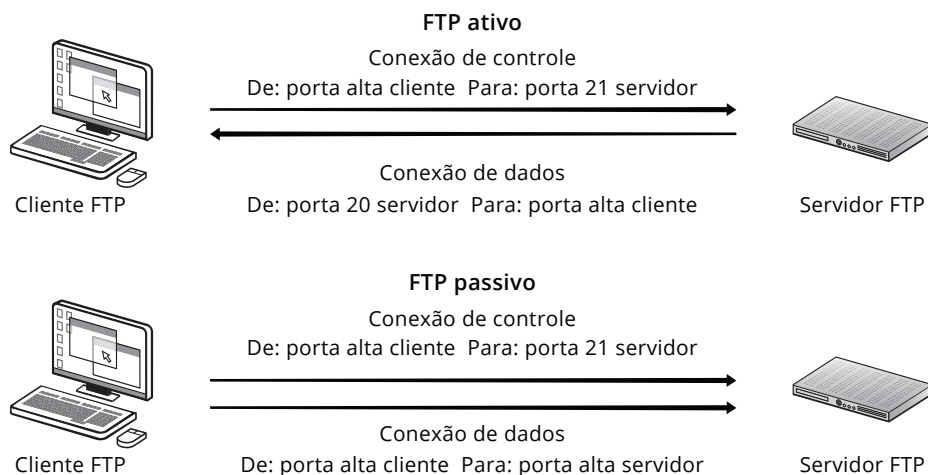


**Figura 5.3**  
Funcionamento do FTP.

O protocolo FTP é baseado em conexões TCP. No entanto, existem dois tipos de conexão:

- **Conexão de controle:** estabelecida e mantida durante todo o tempo de sessão; é por essa conexão que são enviados os comandos;
- **Conexões de dados:** estabelecidas sempre que é solicitada uma transferência de dados. Cada um desses arquivos é transferido em uma conexão separada. Pode haver múltiplas conexões simultâneas.

## Tipos de servidor FTP



**Figura 5.4**  
Tipos de Servidores – Ativo e Passivo.

Um servidor FTP pode funcionar no modo passivo ou ativo. Nos dois casos existem as conexões de controle e as conexões de dados. A diferença está na parte que origina cada conexão.

### Servidor FTP ativo

O cliente se conecta ao servidor por meio de uma conexão de controle, que se origina em uma porta alta do cliente e se destina à porta 21 do servidor. Quando é solicitado o envio de arquivos, o servidor inicia uma conexão de dados que se origina na porta 20 do servidor e se destina a uma porta alta no cliente. Desse modo, a conexão de controle é iniciada pelo cliente, e as conexões de dados são iniciadas pelo servidor.

O acesso a um servidor ativo muitas vezes é bloqueado pelo firewall da rede, uma vez que as conexões de dados são iniciadas por máquinas de fora da rede e com destino a máquinas na rede interna.

! A resposta ao comando `ls`, que lista os diretórios do servidor, é enviada pela porta 20.

## Servidor FTP passivo

O cliente se conecta ao servidor por meio de uma conexão de controle, que se origina em uma porta alta do cliente e se destina à porta 21 do servidor. A diferença é que, quando é solicitada a transferência de arquivos, a conexão de dados é iniciada por uma porta alta do cliente e se destina a uma porta alta no servidor. Desse modo, o servidor nunca inicia uma conexão TCP, que é sempre iniciada pelo cliente. Essa opção funciona melhor quando o cliente está protegido por um firewall.

No Linux o padrão para servidores FTP é o modo passivo.

Configuração:

- Servidor ativado sob demanda no `inetd` (Debian) ou `xinetd` (Red Hat/Fedora).

Arquivos de configuração:

- `ftppaccess`: define autorizações para acesso.
- `ftpconversions`: permite conversão de arquivos.
- `ftpshosts`: permite o controle de acesso a contas no servidor.
- `ftpusers`: lista usuários que não acessam FTP.

Alguns arquivos de configuração devem ser editados antes da inicialização do servidor FTP embutido na instalação no Linux, tais como:

- **ftppaccess**: define autorizações de acesso a arquivos, o número máximo de falhas de login e controles gerais de comportamento do servidor.
- **ftpconversions**: permite a conversão dos arquivos antes que sejam transferidos.
- **ftpshosts**: permite o controle de acesso a contas no servidor, podendo permitir ou negar o acesso a determinados usuários, hosts ou redes.
- **ftpusers**: lista os usuários, normalmente de sistema, que não podem acessar o servidor FTP.

## Estrutura de diretórios típica

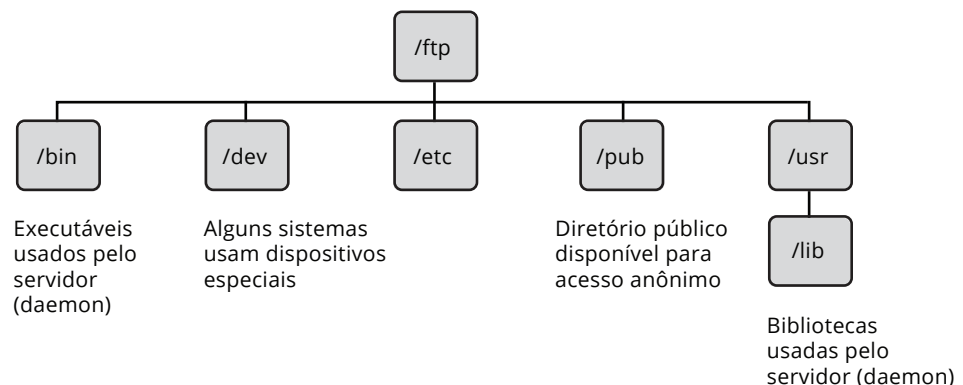


Figura 5.5  
Estrutura de diretórios.

Em uma estrutura de diretórios típica existe o diretório *pub*, além dos diretórios referentes ao binário (daemon) e bibliotecas utilizadas. Esse diretório é indicado como o diretório raiz para usuários anônimos.

Alguns comandos FTP:

- Controle, conexão e sessão.
  - ▣ open, close, quit e bye.
- Envio e recebimento de arquivos:
  - ▣ get, mget, put, mput e binary.
- Diretórios e lista de arquivos.
  - ▣ ls, dir e cd.
- Remoção e criação de arquivos e pastas.
  - ▣ delete, rename, rmdelete e mkdir.



Alguns comandos utilizados para conexão ao servidor FTP e envio ou recebimento de arquivos:

- **?**: ajuda dos comandos.
- **binary**: coloca as transferências em modo binário.
- **bye**: termina a sessão de FTP e sai.
- **cd**: muda de diretório a ser trabalhado.
- **close**: termina apenas sessão do FTP.
- **delete**: remove um arquivo do servidor.
- **dir**: cria uma lista de arquivos do diretório.
- **get**: recebe um arquivo.
- **help**: exibe a ajuda do servidor local.
- **cd**: muda o diretório.
- **ls**: lista o conteúdo dos diretórios.
- **rmdelete**: remove vários arquivos.
- **mget**: recebe vários arquivos.
- **mkdir**: cria um diretório no servidor.
- **mput**: envia vários arquivos.
- **open**: conecta a um servidor FTP.
- **put**: envia um arquivo.
- **pwd**: exibe o nome do diretório atual.
- **quit**: termina a sessão do FTP e sai.
- **recv**: recebe um arquivo.
- **rename**: renomeia um arquivo.

Problemas com firewalls:

- Firewalls normalmente não aceitam conexões de entrada.

Considerações de segurança:

- Restrição de acesso a usuários.
- Envio de arquivo como anônimo (Anonymous Upload).



- Mudar banner de saudação.
- Utilização de SCP como alternativa ao FTP.



## Problemas com firewall

Firewalls normalmente não aceitam conexões de entrada. Desse modo, um cliente inicia uma conexão ao servidor FTP normalmente. Entretanto, ao utilizar os comandos *ls*, *dir* ou *get*, o que inicia uma conexão de dados originada do servidor FTP em direção ao cliente, a transação é bloqueada impedindo a utilização do serviço. Para isso são possíveis duas soluções:

1. Utilizar o servidor FTP em modo ativo: criar regras que permitam o funcionamento das conexões de controle e de dados. Em especial o estabelecimento de uma conexão originada do servidor FTP, caso não seja permitido utilizar servidor passivo.
  2. Utilizar o servidor FTP em modo passivo: criar regras que permitam o estabelecimento de conexões de controle e de dados originadas a partir do cliente.
- **Restrição de acesso a usuários:** adicionar restrição de usuários do sistema que possuem privilégios de acesso e que por isso não podem utilizar o FTP.
  - **Acesso de usuários anônimos:** caso seja permitido o acesso a usuários anônimos, verificar se a pasta pública possui permissão somente de leitura.
  - **Mudar banner de saudação:** trocar o valor da diretiva `ftpd_banner`. O valor padrão traz indicações do sistema, informações que podem ser utilizadas por usuários maliciosos.
  - **Utilização de SCP como alternativa ao FTP:** a adoção de medidas de segurança pode prevenir uma invasão ou utilização indevida do serviço. Todavia, os dados transmitidos, inclusive login e senha, são transmitidos em texto plano. Considere a utilização de Secure Copy (SCP) ou Secure FTP (SFTP) para a transmissão de arquivos sigilosos.

## Secure Shell (SSH)

- Permite a utilização de um terminal remoto e conexões para transferência de arquivos entre clientes e servidores.
- Utiliza autenticação para estabelecer sessão.
- Utiliza encriptação para transmissão de dados.



Secure Shell (SSH) surgiu como uma substituição ao Remote Shell (RSH), que permite conectar um shell a uma máquina remota. O RSH possui dois problemas:

1. Do mesmo modo que o Telnet, todo o tráfego é feito em texto plano, não garantindo a privacidade da informação;
2. Os arquivos `/etc/hosts.equiv` e `~/.rhosts` listam máquinas e usuários autenticados, podendo fazer conexão RSH sem futuras autenticações.

SSH encripta todo o tráfego, incluindo a senha ou chave de autenticação. Também usa chaves de hosts para identificação dos dois hosts envolvidos na comunicação.

A versão do SSH1 era aberta até o release 1.2.12. A partir da versão 1.2.13, a licença mudou para comercial. O OpenSSH tem a licença GPL, e foi desenvolvido a partir do SSH versão 1.2.13, tendo implementado também as características do protocolo SSH2. Se possível, deve-se usar clientes e servidores que suportem SSH2.

## OpenSSH

- Implementação open source do SSH.
- Solução amplamente utilizada em ambiente Linux.
- Existem ainda OpenSSH Secure Copy (SCP) e Secure FTP (SFTP).
  - SCP e SFTP são utilizados como alternativa ao FTP.
- OpenSSH é utilizado como alternativa ao Telnet.
  - A utilização de SSH não é restrito ao mundo Linux.
  - Existem vários clientes Windows para acesso ao SSH e SCP.



OpenSSH é uma implementação open source do SSH. Desde a versão 2.9, usa chaves RSA como padrão. O pacote OpenSSH possui os seguintes componentes:

- **ssh**: cliente SSH (console remoto);
- **sshd**: servidor de shell seguro SSH;
- **scp**: programa para transferência de arquivos entre cliente e servidor;
- **ssh-keygen**: gera chaves de autenticação para o SSH;
- **sftp**: cliente FTP com suporte para comunicação segura;
- **ssh-add**: adiciona chaves de autenticação DSA ou RSA ao programa de autenticação;
- **ssh-agent**: agente de autenticação, sua função é armazenar a chave privada para autenticação via chave pública (DSA ou RSA);
- **ssh-keyscan**: escaneia por chaves públicas de autenticação de hosts especificados. O principal objetivo é ajudar na construção do arquivo local *known\_hosts*.

## Configuração do servidor OpenSSH

- Requer o pacote:
  - openssh-server
- Editar arquivo de configuração:
  - /etc/ssh/sshd\_config
- Ativar o serviço sshd.
- A configuração de um cliente OpenSSH requer os pacotes:
  - openssh-clients
  - openssh



Para utilização do serviço OpenSSH, é necessário que a máquina remota, denominada servidor, tenha o pacote *openssh-server* instalado. É necessário ainda que o arquivo de configuração esteja com a configuração correta, e opcionalmente que o binário *sshd* esteja configurado para iniciar juntamente com o sistema.

Embora o OpenSSH esteja pronto para execução logo após instalado, recomenda-se que alguns ajustes sejam feitos no arquivo de configuração, o que eleva a segurança do sistema. Entre os ajustes possíveis:

- Escolha da porta em que o serviço opera (o padrão é 22);
- Versão do SSH utilizada (o padrão é 2,1);
- Acesso à máquina como root (o padrão é não);

- Ignora arquivos *.rhosts* e *.shosts* (o padrão é sim);
- Permitir senhas em branco (o padrão é não);
- Servidor SFTP: opção padrão é habilitar servidor SFTP.

Logo, recomenda-se:

- Mudar a porta utilizada;
- Quanto à versão, ajustar para aceitar somente a versão SSH2;
- Não permitir acesso como root pelo OpenSSH;
- Ignorar arquivos *.rhosts* e *.shosts*, uma vez que eles autorizam logins subsequentes sem senha;
- Não permitir senhas em branco;
- Habilitar a utilização do servidor SFTP (opcional).

A seguir são apresentados alguns parâmetros do arquivo de configuração.

## Comando *ssh*

Permite efetuar login e executar comandos em uma máquina remota. Exemplo:

- Comando: `# ssh usuario1@leao.linux.lan`
- Efetua conexão à máquina *leao.linux.lan*, autenticando-se como *usuario1*.
- Será solicitada senha.
- Em seguida será concedido acesso à máquina remota.



Parte do conteúdo do arquivo */etc/ssh/ssh\_config*:

```
# /etc/ssh/ssh_config
Port 22
Protocol 2,1
ListenAddress 0.0.0.0
PermitRootLogin no
X11Forwarding no
CheckMail no
RSAAuthentication yes
DSAAuthentication yes
RhostsAuthentication no
RhostsRSAAuthentication no
IgnoreRhosts yes
PasswordAuthentication yes
PermitEmptyPasswords no
```



## Estabelecendo conexão

- Ao conectar-se pela primeira vez a determinado servidor, é exibida a seguinte mensagem:
  - “The authenticity of host ‘leao.linux.lan (192.168.0.3)’ can’t be established.”
  - RSA key fingerprint is 51:2f:c5:95:38:a3:6d:3e:43:ff:1e:ef:4c:f0:c6:01.
  - “Are you sure you want to continue connecting (yes/no)?”
- Respondendo “yes”, a máquina remota é adicionada em uma lista de máquinas conhecidas (known\_hosts).

Ao conectar-se pela primeira vez a determinado servidor, uma mensagem apresenta uma chave e solicita a confirmação de que se trata da máquina que se quer acessar.

“The authenticity of host ‘leao.linux.lan (192.168.0.3)’ can’t be established.”

RSA key fingerprint is 51:2f:c5:95:38:a3:6d:3e:43:ff:1e:ef:4c:f0:c6:01.

“Are you sure you want to continue connecting (yes/no)?”

Respondendo “yes”, a máquina remota é adicionada em uma lista de máquinas conhecidas (known\_hosts). Nas próximas vezes que for feito login nesse mesmo host, apenas a senha será solicitada. Após o estabelecimento da conexão, é solicitada a senha do usuário remoto (root ou outro usuário qualquer).

Em acessos subsequentes, caso seja apresentada uma mensagem de aviso do tipo “Warning: Remote host identification has changed. Someone could be eavesdropping on you right now”, pode estar ocorrendo uma das seguintes situações:

- A máquina remota não é a desejada, sendo no caso uma impostora. Nesse caso não digite nenhuma senha. Convém informar ao administrador da máquina remota.
- Outra situação é aquela em que a máquina remota teve seu sistema ou o OpenSSH reinstalado. Nesse caso a chave pública localizada no arquivo *known\_hosts* deve ser removida conforme instruções na mensagem.

## SCP

- Permite a cópia de arquivos entre hosts de forma segura.
- Utiliza os mesmos mecanismos de segurança do SSH.
- Exemplo de utilização:

```
# scp <usuario-remoto>@<host-remoto>:<diretorio/arquivo> <pasta-local>/<arquivo-local>
# scp usuario1@leao.linux.lan:/tmp/teste.txt
```

- Copia o arquivo *teste.txt* do computador remoto leao.linux.lan para o diretório atual.

O Secure Copy (SCP) é uma alternativa ao FTP na transferência de arquivos e faz parte do pacote OpenSSH. Ao contrário do FTP, que não é um método seguro e cujos dados trafegam em texto plano, no SCP a conexão é autenticada e os dados criptografados. Outra vantagem do SCP é o fato de os comandos serem similares ao comando *cp*. Desse modo, a transferência de um ou mais arquivos entre duas máquinas pode ser feita com um comando do tipo:

```
# scp usuario1@leao.linux.lan:/tmp/teste.txt
```

Podem ainda ser transferidos todos os arquivos de uma determinada pasta:

```
# scp -R usuario1@leao.linux.lan:/home/usuario1/pasta/*
```

R (recursividade): todas as pastas e arquivos a partir da pasta.

Ou ainda na direção contrária:

```
# scp * usuario1@leao.linux.lan:/home/usuario1/
```

## SFTP

- Permite realizar de forma segura a transferência de arquivos entre hosts.
- Utiliza os mesmos mecanismos de segurança do SSH.
- Exemplo de utilização:

```
# sftp usuario@<host-remoto>
```



Assim como o SCP, o SFTP é uma alternativa segura ao FTP na transferência de arquivos e também utiliza criptografia para transmissão de dados. Uma vantagem do SFTP em relação ao SCP ocorre quando não sabemos exatamente a localização do arquivo a ser transferido.

A seguir é apresentado um exemplo de acesso, no qual é efetuado o download do arquivo *teste.txt*:

```
# sftp 192.168.0.1
Connecting to 192.168.0.1
root@192.168.0.1's password:
sftp> ls
pasta1
pasta2
pasta3
sftp> cd pasta1
sftp> ls
teste.txt
sftp> get teste.txt
teste.txt 100% 10KB 12.3KB/s
sftp> exit
#
```

## Geração de chaves

```
ssh-keygen -t <tipo>
```

- Onde <tipo> pode ser:
  - **rsa1**: gera chaves no padrão RSA para a versão SSH1.
  - **rsa**: gera chaves no padrão RSA para a versão SSH2.
  - **das**: gera chaves no padrão DSA apenas para versão SSH2.
  - **ecdsa**: gera chaves no padrão ECDSA apenas para versão SSH2.



Em algumas situações pode ser necessária a utilização de scripts que copiarão arquivos de um servidor para outro, o que geralmente ocorre em situações de replicação de conteúdo entre servidores espelho, de backup, de armazenamento de logs etc. Nesses casos, esses scripts precisarão acessar o servidor remoto para enviar ou buscar arquivos sem a utilização de senha. Para isso podem ser geradas chaves de encriptação baseadas em endereços IP dos dois servidores.



Para evitar que um usuário ou administrador de um servidor acesse livremente o outro, recomenda-se a criação de usuários com poucos privilégios nas duas máquinas.

A seguir, os passos necessários:

- Gerar a chave com o comando `ssh-keygen -t <tipo>`;
- Copiar o arquivo `id_rsa.pub`, se for escolhido o tipo RSA para a máquina de destino;
- No destino deve ser criada uma pasta oculta `.ssh` com permissão 700;
- O usuário deve incluir o arquivo `id_rsa.pub` no arquivo `authorized_keys` na máquina remota.

## Agente SSH

- Permite automatizar a autenticação através de um agente SSH.
- Senha solicitada apenas uma vez em cada sessão Linux.
- As chaves são removidas após o encerramento da sessão.



A automatização do acesso por SSH sem senha pode ser feita com a utilização de um agente SSH. Essa opção deve ser considerada em situações em que o servidor remoto vai ser acessado repetidas vezes ao longo de um período, mas o acesso sem senha não deve ser tornado permanente. Nesse caso usa-se um meio termo no qual a senha será armazenada por determinado Shell, que, enquanto estiver aberto, acessará o servidor sem senha. Para isso são usados dois comandos:

- **ssh-agent \$SHELL**: cria uma sessão com o shell padrão;
- **ssh-add**: acrescenta as chaves ao agente SSH.



# Roteiro de Atividades 5

Nestas atividades os alunos devem trabalhar com três máquinas virtuais:

- Servidor Linux.
- Cliente Linux.
- Cliente Windows.

## Atividade 5.1 – Configure o servidor DHCP

O objetivo do serviço Dynamic Host Configuration Protocol (DHCP) é automatizar a distribuição de endereços e configurações do protocolo TCP/IP para quaisquer dispositivos conectados a uma rede, como computadores, impressoras, hubs e switches.

Para configurar um servidor DHCP no Linux, altere o arquivo */etc/dhcpd.conf*. Esse serviço também usa o arquivo */var/lib/dhcp/dhcpd.leases* para armazenar o banco de dados de endereços “alugados” para clientes.

- Configure o servidor com o endereço IP 192.168.1.1/24 na interface conectada à rede virtual “rede-local”;
- Configure o servidor de DHCP para distribuir os endereços da faixa 192.168.1.10 até 192.168.1.20;
- Teste o funcionamento com o cliente Linux;
- Quais informações podem ser encontradas no arquivo */var/lib/dhcp.leases*?

1. Edite o arquivo de configuração */etc/dhcp/dhcpd.conf* com os parâmetros desejados:

```
#Rede a qual se aplica

subnet 192.168.1.0 netmask 255.255.255.0 {

    # Roteador padrão

    option routers 192.168.1.254;

    # Máscara de rede

    option subnet-mask 255.255.255.0;

    # Configurações de DNS

    option domain-name “empresa.com.br”;

    option domain-name-servers 192.168.1.1;

    # Configurações WINS

    option netbios-name-servers 192.168.1.1;

    # Definição de horário 18000 = Brasil

    option time-offset -18000;

    # Definições do Range IP

    range 192.168.1.10 192.168.1.20;
```



```
default-lease-time 21600;

max-lease-time 43200;

}
```

Reinicie o servidor DHCP:

```
# /etc/init.d/isc-dhcp-server restart
```

1. Verifique o conteúdo do arquivo *dhcpd.leases*;
2. Verifique o conteúdo do arquivo de *dhcpd.lease* antes da inicialização do cliente.
3. O arquivo não deve possuir nenhuma entrada.
4. Reinicie a interface de rede da estação de trabalho Linux.

Verifique se no arquivo de configuração da interface (/etc/network/interfaces) está descrito que será feita pesquisa por um servidor de DHCP. Exemplo:

```
auto eth1
iface eth1 inet dhcp
```

5. Verificar a existência do arquivo *dhcpd.leases*:

```
# find / -name dhcpd.leases

/var/lib/dhcp/dhcpd.leases
```

6. Verifique o conteúdo do arquivo de *dhcpd.lease* após a inicialização do cliente.

O arquivo deve possuir uma entrada para a máquina cliente. Como existirão diversas estações de trabalho configuradas como servidor DHCP, é provável que alguns servidores não consigam distribuir endereços. Isso acontece porque, quando o cliente DHCP envia a requisição, ele receberá o endereço IP do servidor DHCP que responder primeiro.

7. Reinicie o cliente e verifique as alterações no arquivo de *lease*.
8. Reinicie a interface de rede do Windows algumas vezes avaliando as modificações em *dhcpd.leases*.

## Atividade 5.2 – Configuração de um endereço IP fixo

Configure o servidor de DHCP para sempre fornecer o endereço 192.168.1.3 para a interface de rede com o endereço MAC 00:D0:B7:01:D1 (máquina virtual Linux).

## Atividade 5.3 – Utilização de redes compartilhadas

Para realizar esta atividade, é necessário que o servidor esteja configurado com o endereço IP 192.168.1.1 na interface conectada à rede virtual rede-local (exemplo: eth1) e com o endereço IP 192.168.2.1 na interface conectada à rede virtual windows (exemplo: eth2).

A máquina cliente-linux deve ter apenas a conexão com a rede virtual “rede-local”.

A máquina Windows deve ter apenas a conexão com a rede virtual “Windows”.

- Configure o servidor DHCP para fornecer os endereços 192.168.1.100 – 192.168.1.130 na interface eth1;

- Configure o servidor DHCP para fornecer os endereços 192.168.2.30 – 192.168.2.50 na interface eth2;
- Utilize a opção shared-network.

## Atividade 5.4 – Serviço Telnet

---

Todos os servidores DHCP das máquinas virtuais devem estar desativados. Para desativar o serviço DHCP na inicialização do Debian, utilize o binário rcconf. Recomenda-se que todas as estações sejam reiniciadas.

Esta atividade pode ser feita tanto pelo cliente Windows quanto pelo cliente Linux. É importante que tanto a máquina cliente quanto o servidor estejam na mesma rede.

O serviço Telnet permite a um administrador de rede controlar remotamente um equipamento, executando aplicações e comandos como se o teclado de sua estação estivesse conectado diretamente à estação remota.

O protocolo Telnet não realiza a criptografia de tráfego de rede fim a fim, ou seja, um atacante poderia, capturando pacotes de rede, observar todos os comandos enviados pelo administrador para a estação de trabalho remota.

Instale o serviço de Telnet no servidor Linux e teste a conexão a partir do cliente. Lembre-se de que para ativar o serviço do Telnet no Linux será necessário utilizar o xinetd.

## Atividade 5.5 – Serviço FTP

---

Esta atividade pode ser feita tanto pelo cliente Windows quanto pelo cliente Linux. É importante que tanto a máquina cliente quanto o servidor estejam na mesma rede.

O protocolo File Transfer Protocol (FTP) permite a um usuário remoto transferir grandes arquivos para um servidor ou vice-versa.

- Instale e configure o pacote vsftpd no servidor Linux;



Parâmetros importantes: listen, local\_enable e write\_enable.

- Crie um novo usuário chamado usuario1 pertencente ao grupo *users*;
- Utilizando esse usuário, acesse a partir da máquina virtual cliente o serviço de FTP.

## Atividade 5.6 – Criação de uma conexão segura a um host remoto

---

Nesta atividade, os alunos deverão utilizar o cliente e o servidor Linux, verifique se ambos estão na mesma rede, se o servidor possui o endereço 192.168.1.1 e o cliente, o endereço 192.168.1.2.

Devido às vulnerabilidades encontradas no serviço Telnet, foi criado o protocolo SSH, que cifra todo o tráfego de rede gerado entre administrador e estação remota.

- Instale o serviço SSH no servidor.
- A partir do cliente Linux (192.168.1.2), utilize o comando *ssh* para acessar o servidor Linux (192.168.1.1) e em seguida execute o comando *hostname*.



## Atividade 5.7 – Automatizar a criação de conexão segura com o servidor remoto por meio do compartilhamento da chave pública

---

Nesta atividade os alunos deverão utilizar o cliente e o servidor Linux, verifique se ambos estão na mesma rede e se o servidor possui o endereço 192.168.1.1 e o cliente possui o endereço 192.168.1.2.

Escolha uma única estação para gerar a chave a ser utilizada para a autenticação.

O objetivo é estar apto a configurar um servidor para permitir acesso SSH sem senha para uma máquina cliente.

- Utilizando o `usuario1` criado na Atividade 5.5, configure o acesso SSH a partir do cliente Linux para o servidor Linux.
- Será necessário criar uma chave pública para o `usuario1` na máquina cliente (veja o manual do comando `ssh-keygen`).

## Atividade 5.8 – Acesso de grupos a diretório compartilhado no FTP

---

Esta atividade pode ser feita tanto pelo cliente Windows quanto pelo cliente Linux. É importante que tanto a máquina cliente quanto o servidor estejam na mesma rede.

Em algumas situações é interessante compartilhar um mesmo diretório para diversos usuários do FTP.

- Crie o grupo `usuarios-ftp` no LDAP;
- Crie o diretório `/home/usuarios-ftp` e conceda as devidas permissões ao grupo;
- Crie os usuários `usuarioftp1` e `usuarioftp2` e defina o diretório `/home/usuarios-ftp` como seu `homedir`;
- Inclua os usuários `usuarioftp1` e `usuarioftp2` no grupo `usuarios-ftp`;
- Testar o funcionamento.



Atenção para os parâmetros do arquivo `/etc/vsftpd.conf` `local_enable`, `write_enable` e `local_umask`.

## Atividade 5.9 – Transferência de um arquivo de um host remoto para o host local e vice-versa

---

Nesta atividade os alunos deverão utilizar o cliente e o servidor Linux. Verifique se ambos estão na mesma rede e se o servidor possui o endereço 192.168.1.1 e o cliente possui o endereço 192.168.1.2.

1. Utilize o comando `scp` para transferir um arquivo do cliente Linux para o servidor.

## Atividade 5.10 – Criando uma conexão FTP segura com o comando SFTP

---

Nesta atividade os alunos deverão utilizar o cliente e o servidor Linux.

1. Verifique se ambos estão na mesma rede e se o servidor possui o endereço 192.168.1.1 e o cliente possui o endereço 192.168.1.2.
2. Utilize o comando SFTP para transferir um arquivo do servidor para o cliente Linux.

# 6

## Servidor web

### objetivos

Instalar e configurar o servidor web Apache, em versão compilada e através do código-fonte, configurar servidores virtuais e o servidor web seguro, usando certificados digitais, habilitar no Apache o recurso de publicação de páginas pessoais.

Fundamentos de servidores web: arquitetura, esquema de funcionamento, protocolo HTTP, servidor Apache, recursos comuns a servidores web como proxy web, domínio virtual e servidor seguro (SSL).

### conceitos

### Introdução

Este capítulo aborda os conceitos fundamentais de servidores web, como arquitetura, esquema de funcionamento, protocolo HTTP, servidores web, servidor Apache e a instalação e testes com o servidor Apache.

Na primeira parte são apresentados os conceitos iniciais, arquitetura e protocolo. Em seguida são abordados os recursos comuns aos diversos servidores web atualmente disponíveis, tais como:

- Proxy web;
- Domínio virtual;
- Servidor seguro (SSL).

Por último são apresentados os recursos do servidor web Apache, inclusive sua instalação, configuração e testes.

### Conceitos fundamentais

World Wide Web (WWW):

- Desenvolvida para permitir o acesso a informações organizadas em forma de hipertexto.
- Permite a recuperação de informação:
  - Texto.
  - Imagens (estáticas e animadas).
  - Áudio etc.
- Baseada na arquitetura cliente/servidor.





- O servidor armazena um conjunto de arquivos de dados que são transferidos para o cliente.

Identificação de servidores, serviços e arquivos de modo uniforme.

- Uniform Resource Locator (URL).

A internet, muitas vezes denominada World Wide Web (WWW), permite um grande e variado trânsito de informações, tais como textos, documentos, conteúdos multimídia, mensagens etc.

Uma das ferramentas mais utilizadas para o envio e a recuperação de informações são os servidores web, que permitem o envio de textos, mensagens, imagens e áudio. Para isso, um determinado servidor web pode usar páginas estáticas, que são nada mais do que arquivos presentes no sistema de arquivos do servidor, ou páginas dinâmicas, que são conteúdos gerados dinamicamente e sob demanda para o cliente que as solicita. Essa última forma é muito utilizada em transações pela internet. Nesse caso, o conteúdo pode ser gerado pelo próprio servidor web que atende ao pedido dos clientes, ou então por outro servidor acoplado que processa e repassa o conteúdo.

A comunicação web lida com um processo do tipo cliente/servidor. Nesse caso um cliente (navegador web) envia uma solicitação ao servidor web, que processa a solicitação e envia a resposta ao cliente. A solicitação enviada pelo cliente consiste em uma Uniform Resource Locator (URL), por exemplo: <http://www.rnp.br>. Essa solicitação é enviada para o servidor web, que encaminha o conteúdo associado a esse endereço, que pode conter textos, imagens e áudio.

A sigla URL era anteriormente conhecida por Universal Resource Locator.

## Campos da URL

Protocolo: // servidor [:porta] / caminho / arquivo

- Indica o serviço desejado:
  - Hyper Text Transfer Protocol (HTTP).
  - File Transfer Protocol (FTP).
  - MAILTO (para correio eletrônico).
  - Terminal Network (Telnet).
  - File (para arquivo local).

A solicitação enviada pelo cliente consiste em uma URL, que possui componentes como: protocolo, servidor, porta, caminho e arquivo. Essas informações servem para designar o serviço, protocolo e porta que devem ser acessados e utilizados na obtenção dos dados.

Desse modo, mesmo quando vários serviços diferentes estão disponíveis no servidor em questão, cada um pode trabalhar de modo independente do outro. Logo poderemos ter em uma mesma máquina diversos serviços, como servidor web, servidor FTP, servidor DNS etc.

Servidor [:porta]

- Indica o endereço do servidor desejado.
- Pode ser um Fully Qualified Domain Name (FQDN): nome da máquina, domínio e domínio de topo.
- Ou somente o nome da máquina para acessos a partir da mesma rede.





Caminho:

- Indica o caminho no sistema de arquivos do servidor web.

Arquivo:

- Indica o arquivo desejado.

Um servidor web pode possuir grande quantidade de informação, dispersa por uma grande quantidade de arquivos e pastas. Por isso, é necessário que o cliente, ao buscar por determinado dado, informe o caminho completo até ele, o que usualmente é feito por meio da digitação da URL no navegador, ou de um clique em determinado link disponível em determinada página da internet.

O caminho indica em qual pasta ou arquivo a partir de uma pasta raiz está o dado procurado. Essa pasta raiz normalmente não coincide com a raiz do sistema no qual o servidor web está instalado.

### Exemplo de URL

Acesso à página principal da RNP.

- http://www.rnp.br:80/index.html
  - Protocolo: http
  - Servidor: www.rnp.br
  - Porta: 80
  - Caminho: /
  - Arquivo: index.html

Ou simplesmente:

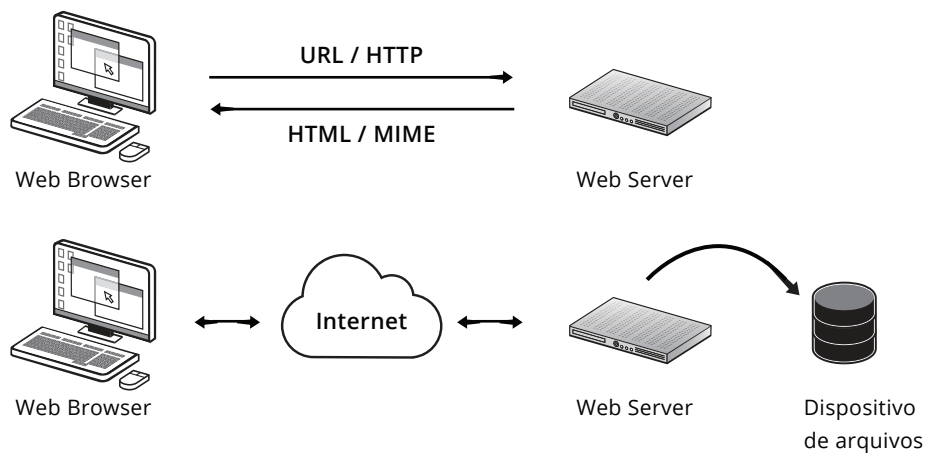
- www.rnp.br



Entre as várias informações de uma URL, usamos: protocolo, servidor, porta, caminho e arquivo. Alguns desses dados podem ser omitidos quando digitamos a URL no navegador ou em um link da página web. Logo, o acesso a determinada página web pode ser feito digitando, por exemplo: http://www.rnp.br:80/index.html ou www.rnp.br

No exemplo apresentado, não foi necessário, ao informar uma URL no navegador, passar o protocolo, porta e arquivo. Isso é possível porque os navegadores web encaminham por padrão uma consulta utilizando o protocolo HTTP e a porta padrão 80. O servidor web, por sua vez, envia como resposta a página padrão especificada em seu arquivo de configuração, nesse caso, *index.html*.

## Esquema de funcionamento



**Figura 6.1**  
Esquema de funcionamento: transferência de dados entre cliente (browser) e servidor.

### Servidor web

- Todos os web sites têm processos servidores que escutam a porta TCP 80, aguardando conexões dos clientes.
- Após estabelecida uma conexão, o cliente envia uma solicitação e o servidor envia uma resposta.
- O protocolo que define as solicitações e respostas válidas é chamado de HyperText Transfer Protocol (HTTP).

### Cliente web

- Disponibiliza um mecanismo de busca, transferência e apresentação de dados contidos em servidores web.
- O navegador, também denominado browser, busca a página solicitada, interpreta seu texto, seus comandos de formatação e apresenta os dados na tela.
- Entre os navegadores mais conhecidos hoje estão: Netscape, Mozilla, Firefox, Thunderbird, Konqueror, Internet Explorer, Safari. Além do Lynx, um navegador de linha de comando muito utilizado em testes e/ou em computadores sem a interface gráfica instalada.

## Protocolo HTTP

Hyper Text Transfer Protocol (HTTP).

- Protocolo utilizado para a transferência de informações na web, utilizando TCP.
- Interação consiste em um pedido ASCII, seguido por uma resposta Multipurpose Internet Mail Extensions (MIME).
  - MIME é uma forma padronizada para o envio de mensagens de diferentes tipos e codificações.
  - Tipos: texto, imagens, vídeo, áudio, aplicações etc.

O protocolo HTTP é um protocolo de camada de aplicação utilizado em sistemas distribuídos, colaborativos e de hipermídia. Sua primeira versão HTTP/0.9 está em uso desde 1990, sendo até então um protocolo simples para transferência de dados brutos pela internet.



A versão HTTP/1.0 melhorou o protocolo ao permitir a transferência de mensagens no formato MIME, contendo metadados e modificadores nas solicitações e respostas. No entanto, não suportava caching, proxies hierárquicos, conexões persistentes ou hosts virtuais.

A versão HTTP/1.1 foi desenvolvida para atender a essas necessidades, e ainda às necessidades adicionais de buscas, atualizações de front-end, incluindo um amplo conjunto de métodos e cabeçalhos que indicam a finalidade da solicitação. As mensagens são passadas em um formato semelhante ao das mensagens de e-mail, utilizando o formato MIME.

O HTTP é ainda um protocolo genérico para outros sistemas internet, como SMTP, NNTP e FTP, permitindo que diversas aplicações possam acessar conteúdo hipermídia.

## Tipos de pedidos

### Pedidos simples

- Pedido:
  - ▣ GET e página desejada sem a versão do produto.
- Resposta:
  - ▣ Página sem cabeçalho, MIME e codificação.



O protocolo HTTP é um protocolo do tipo pedido/resposta. O cliente envia um pedido ao servidor na forma de um método de requisição, Universal Resource Identifiers (URI), versão do protocolo e uma mensagem em formato tipo MIME contendo modificadores de requisição, informações do cliente e conteúdo.

A resposta do servidor é enviada com uma linha de status que inclui a versão do protocolo, um código de sucesso ou erro seguido de uma mensagem do tipo MIME, que contém informações do servidor, metadados e o corpo da mensagem.

### Pedidos completos

- Pedido:
  - ▣ Versão do protocolo, comando (método utilizado), página desejada, versão do protocolo e cabeçalhos.
- Resposta:
  - ▣ Página com cabeçalho.
- Respostas com códigos de três dígitos:
  - ▣ 1xx: informação.
  - ▣ 2xx: sucesso.
  - ▣ 3xx: redirecionamento.
  - ▣ 4xx: erro Cliente.
  - ▣ 5xx: erro Servidor.



Quando um pedido é enviado, uma resposta retorna contendo um código e o conteúdo solicitado. O código retornado pode indicar sucesso ou fracasso da operação.

Alguns códigos possíveis:

- 100: Continue.
- 101: Switching Protocols.



- 200: Ok.
- 201: Created.
- 202: Accepted.
- 203: Non-Authoritative Information.
- 204: No Content.
- 205: Reset Content.
- 206: Partial Content.
- 300: Multiple Choices.
- 301: Moved Permanently.
- 302: Found.
- 303: See Other.
- 304: Not Modified.
- 305: Use Proxy.
- 307: Temporary Redirect.
- 400: Bad Request.
- 401: Unauthorized.
- 402: Payment Required.
- 403: Forbidden.
- 404: Not Found.
- 405: Method Not Accepted.
- 406: Not Acceptable.
- 407: Proxy Authentication Required.
- 408: Request Time-out.
- 409: Conflict.
- 410: Gone.
- 411: Length Required.
- 412: Precondition Failed.
- 413: Request Entity Too Large.
- 414: Request URI Too Long.
- 500: Internal Server Error.
- 501: Not Implemented.
- 502: Bad Gateway.
- 503: Service Unavailable.
- 504: Gateway Time-out.
- 505: HTTP Version Not Supported.

## Métodos

- GET: pedido para ler uma página.
- HEAD: pedido para ler o cabeçalho de uma página web.
- PUT: pedido para gravar uma página web.

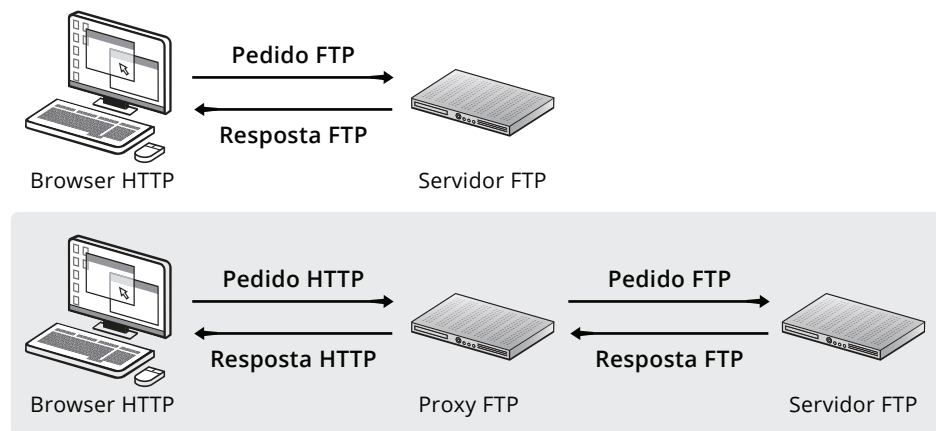


- POST: inserção de um recurso.
- DELETE: remove uma página web.
- LINK / UNLINK – Conecta /: quebra a conexão existente entre dois recursos.

Os métodos são utilizados tanto para solicitações de páginas web quanto para o envio de informações inseridas pelos usuários nessas páginas. Por exemplo: método GET para solicitar uma determinada página web; método PUT para gravar uma página web; método POST para inserir um recurso.

## Servidor web – proxy

Servidores proxy: usuários podem configurar seus browsers com proxies para protocolos que não são capazes de reconhecer.



**Figura 6.2**  
Esquema de funcionamento do servidor web-proxy.

O servidor proxy pode ser um programa executando na mesma máquina do navegador ou pode estar em uma outra máquina em algum ponto da rede, servindo a outros navegadores.

Os usuários podem ainda configurar seus navegadores com proxies para protocolos que os navegadores não são capazes de reconhecer. A utilização de servidores proxy tem ainda como objetivos:

- **Cache:** atuação como servidor cache, aliviando o tráfego em determinados links e agilizando a resposta.
- **Controle de conteúdo:** atua em conjunto com um firewall para limitar os dados que podem trafegar na rede, restringindo determinados protocolos.

## Domínio Virtual

- Diferentes sites são hospedados em um mesmo servidor web.
  - Exemplos: www.exemplo.com e www.exemplo.org
- Utilizado por grandes empresas e/ou provedores de internet.
- Elimina a necessidade de múltiplos servidores.
- Facilitando a administração do servidor web.
- Economia de recursos.
- Hardware, software e infraestrutura.

Domínio Virtual (Virtual Hosting) é uma técnica que permite a hospedagem de múltiplos sites no mesmo servidor, com cada um deles contendo páginas e conteúdos diversos. Esse recurso é amplamente utilizado por grandes empresas, provedores de internet, datacenters etc. Sua grande vantagem é a eliminação da necessidade de servidores dedicados a cada um dos sites que hospeda, permitindo que uma grande empresa que possua sites diferentes associados à cada divisão e/ou produto possa agrupá-los num único servidor, o mesmo valendo para provedores de internet e datacenters, que podem reunir em uma única máquina os diversos clientes que possui.

Vantagens:

- **Facilidade de administração:** em virtude da redução da quantidade de servidores a administrar, requer menor quantidade de hardware e software, o que facilita a aquisição, configuração e backup.
- **Economia de recursos:** menor demanda de recursos de infraestrutura, como espaço físico e consumo de energia.



Quando é utilizado Domínio Virtual, deve-se ficar atento à necessidade de o servidor DNS resolver cada um dos endereços utilizados, para isso devendo possuir um alias (CNAME) para cada site hospedado.

## Secure Socket Layer (SSL)

- Camada de conexão segura.
- Método de criptografia desenvolvido pela Netscape.
- Suporta diversos protocolos de criptografia.
- Fornece autenticação entre o servidor e o cliente.
  - Garantida por uma terceira parte denominada Certification Authority (CA) ou simplesmente Autoridade Certificadora.
  - Verisign, Tawte e CAcert.
  - ICPBrasil: PR, Serpro, Serasa, Certisign, SRF e CEF.



Para a disponibilização de uma conexão segura entre o cliente (navegador) e o servidor web é necessária a utilização de dois recursos:

- **Criptografia:** para evitar que um elemento intermediário na rede tome conhecimento do conteúdo trafegado;
- **Autenticação:** para garantir que a página acessada seja realmente a que se deseja, e não uma página impostora.

O protocolo Secure Socket Layer (SSL) ou camada de conexão segura provê uma criptografia fim-a-fim entre o cliente e o servidor. Suporta diversos protocolos e foi desenvolvido inicialmente pela Netscape para facilitar transações na internet.

O funcionamento se dá em duas etapas:

- **Autenticação:** o cliente solicita a uma entidade certificadora, entre as várias disponíveis em sua relação, que autentique o servidor que se deseja acessar. O cliente pode ou não ser autenticado.
- **Criptografia:** o cliente e o servidor escolhem um protocolo de criptografia comum; a partir de então a comunicação é considerada segura.

## Business-to-Consumer (B2C)

Transação entre empresa e consumidor, utilizada em compras ou transações bancárias pela internet.

Em geral, o cliente procura autenticar o servidor, o que acontece frequentemente em transações

## Business-to-Consumer (B2C).

Os casos em que o servidor procura autenticar o cliente são mais raros; entre eles, podemos citar a declaração de impostos pela internet, por meio de certificados digitais, para comprovar a identidade das partes envolvidas.

## Servidores web

Existe atualmente grande quantidade de navegadores (browsers) de internet.

Considerando os principais que possuem capacidade para executar em ambiente Linux/Unix:

- Código livre e licença GPL (ou similar):
  - Apache HTTP Server, Apache Tomcat Server, AOL Server, JigSaw e Tux Web Server.
- Código fechado e licença proprietária:
  - IBM HTTP Server, Sun Java System Web Server e Zeus Web Server.

Ao analisar os servidores web atualmente disponíveis, podemos considerar algumas características associadas a recursos e disponibilidade.

Em relação aos recursos disponíveis, podemos considerar:

- **Autenticação:** autentica um usuário por meio de um login e senha, com texto plano ou criptografado (HMAC);
- **Criptografia** (SSL e TLS – Transport Layer Security): suporte à conexão encriptada;
- **Domínio virtual:** suporte a múltiplos sites com um único endereço IP;
- **Conteúdo dinâmico** (cgi-bin, Servlet, SSI, PHP e ASP): suporte à geração de páginas interativamente;
- **Módulos** (carregados dinamicamente): capacidade de carregar módulos sob demanda;
- **Compressão de conteúdo:** capacidade de comprimir conteúdo enviado ao usuário para economizar largura de banda;
- **Limitação de usuários e/ou largura de banda:** capacidade de limitar largura de banda e/ou número de conexões inclusive por usuário, evitando saturar o servidor ou a rede;
- **Modo de execução** (espaço de usuário ou núcleo do sistema): forma de acesso aos recursos do sistema, especialmente memória e processador.

Considerando a disponibilidade:

- Sistemas Operacionais em que pode ser executado;
- Código livre: código-fonte disponível ou não;
- Tipo de licença: tipo de licença disponibilizada;
- Custo de aquisição.

## Servidor web Apache

Considerado um dos mais populares servidores HTTP de código aberto. Possui todos os recursos mais utilizados, como:

- Autenticação: texto plano ou criptografado (HMAC).
- Criptografia: SSL e TLS.
- Domínio virtual.





- Conteúdo dinâmico: cgi-bin, Servlet, SSI, PHP, ASP, entre outros.
- Módulos: capacidade de carregar módulos sob demanda.
- Compressão de conteúdo: economia de largura de banda.
- Limitação de usuários e/ou largura de banda: limitar largura de banda e/ou número de conexões, inclusive por usuário.
- Modo de execução: espaço de usuário.



O servidor web Apache contava, em novembro de 2011, com 65% do percentual de sites ativos na internet (conforme pesquisa da Netcraft Web Server Survey – <http://netcraft.com>), tendo seus concorrentes, Microsoft, Nginx e Google, respectivamente 15,66%, 8,54% e 3,47%. Enquanto servidor padrão da internet, o servidor Apache suporta todos os recursos anteriormente mencionados e utilizados atualmente em grande parte dos sites, como: autenticação, criptografia, domínio virtual, conteúdo dinâmico, carga de módulos sob demanda, compressão de conteúdo e limitação de acesso. Compatível com os protocolos padrão de internet, incluindo o HTTP/1.1 (RFC 2616).

- Instalação relativamente simples.
- Composto de daemon que atua na porta 80 (http) e 443 (https), ou qualquer outra porta a seguir da 1024.
- Somente o root pode executar ou parar o servidor.
  - ▣ Ativa vários processos filho que escutarão e responderão às solicitações.
- Mantém vários processos em execução, com múltiplas threads em cada um.
  - ▣ Múltiplos processos: estabilidade.
  - ▣ Múltiplas threads: maior capacidade em atender solicitações, consumindo menos recursos.



O servidor web Apache possui instalação relativamente simples, podendo ser instalado a partir de pacotes prontos ou ser compilado durante o processo de instalação.

Para ser colocado em execução, deve ser ativado o daemon, o que pode ser feito somente pelo root. Esse daemon executa algumas tarefas iniciais básicas, como ativar os arquivos de log e em seguida ativar os processos filhos, que escutarão e responderão às solicitações dos clientes (browsers). O processo httpd principal executa com privilégio de root, enquanto os demais executam com menos privilégios.

O Apache conta com dois tipos de módulos: módulos core, de multiprocessamento e demais módulos.

### Módulo core e de multiprocessamento

Core são recursos sempre disponíveis para suporte a múltiplos módulos de multiprocessamento, inclusive para diversos Sistemas Operacionais. Outros módulos:

- mod\_auth\_basic e mod\_auth\_digest: autenticação simples e MD5;
- mod\_cache: cache de conteúdo baseado em URI;
- mod\_cgid: execução de scripts CGI;
- mod\_dbd: controla conexões com bancos de dados SQL;
- mod\_filter: permite configuração contextualizada de filtro de conteúdo;
- mod\_headers: permite a reconfiguração de cabeçalhos;



- mod\_proxy: implementa um proxy ou gateway para o servidor Apache;
- mod\_proxy\_balancer: permite operações de balanceamento de carga;
- mod\_ssl: suporte à criptografia SSL e TLS.

## Exercício de fixação 1

### Correlacione as colunas

Correlacione as colunas de acordo com as diretivas do Apache.

- |                   |                                                                                                                         |
|-------------------|-------------------------------------------------------------------------------------------------------------------------|
| 1. ErrorLog       | <input type="checkbox"/> Modifica o usuário e o grupo que executarão o Apache.                                          |
| 2. LogLevel       | <input type="checkbox"/> Define a localização e o formato do arquivo de log de acessos.                                 |
| 3. CustomLog      | <input type="checkbox"/> Permite associar o Apache a um IP específico e/ou portas, além do default.                     |
| 4. AccessFileName | <input type="checkbox"/> Define a localização do arquivo de log de erros.                                               |
| 5. Listen         | <input type="checkbox"/> Define o diretório raiz de um domínio.                                                         |
| 6. User/Group     | <input type="checkbox"/> Define o nível de log do servidor.                                                             |
| 7. ServerAdmin    | <input type="checkbox"/> Define o diretório onde serão armazenados os arquivos de configuração do servidor, erro e log. |
| 8. ServerRoot     | <input type="checkbox"/> Permite definir um nome enviado de volta aos clientes.                                         |
| 9. ServerName     | <input type="checkbox"/> Define o nome do arquivo utilizado para controlar o acesso às páginas.                         |
| 10. DocumentRoot  | <input type="checkbox"/> Define o endereço de e-mail para o qual o servidor httpd envia mensagens quando há erros.      |





# Roteiro de Atividades 6

## Atividade 6.1 – Instalação do servidor web Apache

Esta atividade pode ser feita tanto pelo cliente Windows quanto pelo cliente Linux. É importante que tanto a máquina cliente quanto o servidor estejam na mesma rede.

### Instalação através do repositório Debian

1. Execute os comandos:

```
# apt-get update  
# apt-get install apache2
```

2. Para testar a instalação, abra o navegador da estação de trabalho Windows ou Linux e acesse a URL correspondente ao endereço IP do servidor (exemplo: <http://192.168.1.1>).

## Atividade 6.2 – Configuração de servidores virtuais

Ao utilizar servidores virtuais, podem existir duas situações:

- Múltiplos sites para um único endereço IP;
- Múltiplos sites, cada um com um determinado IP.

O primeiro caso é o mais usual, e será praticado nesta atividade. O segundo caso envolve suporte a múltiplos endereços físicos e não será considerado nesta atividade.

No Apache, a partir da versão 2.0, as configurações dos domínios virtuais disponíveis estão no diretório */etc/apache2/sites-available* e os sites habilitados no diretório */etc/apache2/sites-enabled*. Para evitar possíveis diferenças entre esses dois arquivos, é adotada a solução de criar um link simbólico no diretório */etc/apache2/sites-enabled*, apontando para o arquivo no diretório */etc/apache2/sites-available*. Esse procedimento é realizado pelo comando *a2ensite*.



Lembre-se de associar o endereço IP do servidor web no servidor DNS de sua rede para responder a solicitações destinadas ao site meusite.empresa.com.br.

- Crie uma entrada no servidor de DNS para o domínio *meusite.empresa.com.br*;
- Configure um virtual host no servidor Apache para o site *meusite.empresa.com.br*;
- Crie o diretório */var/www/meusite*, que será usado como diretório root do virtual host;
- Crie o arquivo *index.html* com um texto de boas-vindas, para ser exibido quando o domínio for acessado.

## Atividade 6.3 – Configuração de um servidor seguro

O protocolo HTTP não possui nenhum recurso de criptografia e, por consequência, todo o tráfego de rede gerado entre cliente e servidor poderia ser visualizado por um atacante. Para aumentar a segurança de aplicações web, é interessante habilitar o suporte a conexões cifradas através do SSL.

- Instale o OpenSSL;
- Crie um certificado digital auto assinado para o seu site;



```
# openssl req -new -x509 -days 365 -nodes -out /etc/apache2/apache.  
pem -keyout /etc/apache2/apache.pem
```

- Habilite o módulo SSL no Apache (veja o comando *a2enmod*);
- Configure o virtual host do site *meusite.empresa.com.br* para utilizar o protocolo HTTPS.

### Atividade 6.4 – Usando *.htaccess* para definir permissão de acesso a um site

---

O arquivo *.htaccess* permite que sejam feitas modificações na configuração do servidor por diretório. Esse arquivo pode conter uma ou várias diretrizes de configuração, que serão aplicadas ao diretório atual e a todos os seus subdiretórios.

- Crie um diretório chamado “restrito” dentro do diretório root do virtual host *meusite.empresa.com.br*.
- Configure o servidor Apache para que aceite a autenticação de usuários válidos no LDAP quando tentarem acessar o endereço *meusite.empresa.com.br/restrito*.



Para permitir autenticação pelo LDAP é necessário instalar o módulo “libapache-authznetldap-perl”.

- Teste o funcionamento.

Será necessário alterar a configuração do VirtualHost. Pesquise a diretiva AllowOverride.

### Atividade 6.5 – Habilitando páginas pessoais de usuários

---

O módulo *userdir* habilita um usuário a ter seu site dentro da sua pasta pessoal.

- Habilite o módulo *userdir* no Apache usando o comando *a2enmod*;
- Pesquise sobre as configurações do módulo *userdir* e faça as configurações necessárias para que os novos usuários possam ter acesso às páginas pessoais;
- Adicione o usuário *webuser*;
- Crie uma página de boas-vindas e teste o funcionamento.

# 7

## Servidor de correio eletrônico (parte 1)

### objetivos

Projetar, instalar e configurar um servidor de correio eletrônico Postfix, entender o funcionamento do correio eletrônico e dos protocolos SMTP, POP e IMAP e conhecer os recursos da ferramenta Postfix.

Servidor Postfix, teste do servidor com Telnet e envio de mensagem para usuários através do shell de comando.

### conceitos

### Introdução

O estudo deste capítulo tem como objetivo capacitar o aluno a projetar, instalar e configurar um servidor de correio eletrônico Postfix. Serão apresentados conceitos teóricos, o funcionamento do correio eletrônico e os protocolos SMTP, POP e IMAP. Em seguida, aspectos e recursos da ferramenta Postfix. Por último, veremos a parte prática envolvendo instalação, configuração e testes com o servidor de correio eletrônico Postfix, no qual são incentivadas boas práticas de configuração e administração.

### Correio eletrônico

- Também denominado e-mail (eletronic mail).
- Desenvolvido para permitir a troca de mensagens entre usuários por meio eletrônico.
- Permite o envio de arquivos anexados às mensagens.
- Documentos, planilhas, imagens, vídeos etc.



**Figura 7.1**  
Troca de mensagem.



O correio eletrônico é um dos serviços mais usados na internet, permitindo que usuários conectados em qualquer ponto da rede troquem mensagens entre si e com outros. Através dessas mensagens podem ser enviados textos e arquivos anexados, como documentos, planilhas, imagens, vídeos e outros. É uma ferramenta extremamente importante no incremento da produtividade, pois integra e aproxima os usuários com a facilitação da troca de informações.



## Origens do correio eletrônico

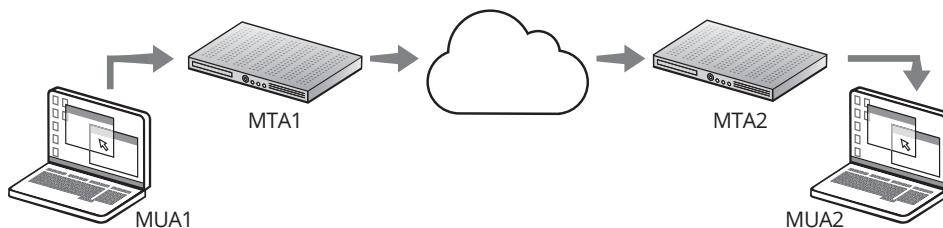
- Criado originalmente em 1965 para a troca de mensagens entre usuários de mainframes.
- Em 1966, já permitia a troca de mensagens entre diferentes computadores.
- Em 1971 surgiu o símbolo @.
- Com o surgimento da internet, seu uso foi multiplicado.

O correio eletrônico foi criado em 1965, originalmente para troca de mensagens entre usuários de computadores de grande porte. Logo foi adaptado para troca de mensagens por meio de terminais remotos. Em 1969, já com a ARPANET, surgiu a utilização do símbolo @ como separador entre nome de usuário e domínio. Com a internet, a utilização do e-mail foi multiplicada, tornando-se juntamente com o acesso à página web um dos serviços mais utilizados.

## Funcionamento do correio eletrônico

Baseado em dois subsistemas:

- Agente de transferência de mensagens – Mail Transport Agent (MTA).
- Agente de usuário – Mail User Agent (MUA).



**Figura 7.2**  
Funcionamento do correio eletrônico.

Durante o processo de envio de mensagens são utilizados dois tipos de programas: de transporte e de usuário. O primeiro tipo cuida do envio da mensagem entre a origem e o destino. O segundo permite ao usuário ler e escrever suas mensagens. Também denominados respectivamente de agentes de transferência de mensagens e agentes de usuário.

Agentes de transferência de mensagens ou Mail Transfer Agent (MTA):

- Permitem o deslocamento das mensagens da origem ao destino;
- São os programas (daemons) que executam no ISP (provedor de internet);
- Exemplo: Sendmail, Postfix, Qmail e Exlm.

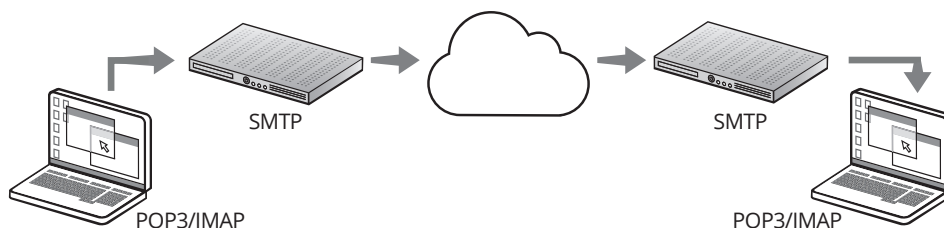
Agentes de usuário ou Mail User Agent (MUA):

- Permitem aos usuários ler e escrever suas mensagens;
- São os programas de e-mail no computador do usuário. Exemplos: Eudora, Evolution, KMail, Outlook e Mozilla Thunderbird.

Protocolos utilizados no envio de mensagens:

- Simple Mail Transfer Protocol (SMTP).
- Post Office Protocol (POP) ou Internet Message Access Protocol (IMAP).

**Figura 7.3**  
Protocolos utilizados.



Ao longo do processo de envio de uma mensagem, são utilizados alguns protocolos, tais como: Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP3) e Internet Message Access Protocol (IMAP). O primeiro é utilizado na transferência de mensagens entre os MTAs, enquanto os outros dois podem ser utilizados no envio de mensagens do MUA para o MTA e vice-versa.

O usuário, ao escrever sua mensagem, pode fazê-lo desconectado da rede (off-line). Para enviar suas mensagens, deve estar conectado à rede, seja ela local ou discada. De maneira simplificada, podemos afirmar que as mensagens são colocadas em uma espécie de caixa de entrada do servidor de correio eletrônico.

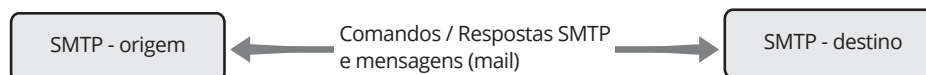
O servidor de correio eletrônico fica constantemente verificando se existem mensagens a serem enviadas. Caso a mensagem seja destinada a usuários no mesmo servidor, é encaminhada a uma caixa de entrada para que sejam distribuídas para os usuários. Se a mensagem for destinada a outro servidor de correio, é solicitada uma conexão com esse servidor, sendo então encaminhada a mensagem a ele, que analogamente coloca a mensagem em uma caixa de entrada para posterior distribuição aos seus usuários.

## Protocolo SMTP

- O objetivo do protocolo SMTP é transferir mensagens de maneira confiável e eficiente.
- Independe do subsistema de transmissão e requer somente um meio de transmissão confiável.



**Figura 7.4**  
Protocolo SMTP.



O protocolo SMTP objetiva transferir mensagens de maneira confiável e eficiente, sendo independente do subsistema de transmissão, requerendo somente um meio de transmissão confiável. É orientado à conexão e utiliza o protocolo Transmission Control Protocol (TCP).

O SMTP funciona do seguinte modo: como resultado da requisição do usuário, o SMTP-origem estabelece uma conexão com o SMTP-destino. O SMTP-destino pode tanto ser o destino final, quanto um intermediário. Os comandos são gerados pelo SMTP-origem e enviados ao SMTP-destino. As respostas fazem o caminho inverso.

Com a conexão estabelecida, o SMTP-origem envia um comando MAIL indicando a origem da mensagem. Caso o SMTP-destino aceite, é retornada uma mensagem "OK" (código 250). O SMTP-origem envia então o destinatário da mensagem. Duas situações podem ocorrer: na primeira, se o destinatário não existir no SMTP-destino, é enviada uma mensagem de ERRO (código 550) informando que o usuário não existe. Na segunda, se o usuário existe, é enviada uma mensagem "OK" (código 250). Múltiplos destinatários podem ser aceitos. Em seguida o SMTP-origem envia o comando DATA. O SMTP-origem informa o código 354 (iniciar mensagem) e informa que a mensagem deve terminar com uma linha constando somente um ponto (<CRLF>.<CRLF>). Após o envio dos dados, o SMTP-destino confirma com OK.



Para o estabelecimento da conexão, o SMTP-origem envia uma mensagem no seguinte formato:

```
SMTP-destino (R): 220 <identificação SMTP-destino> Service ready
SMTP-origem (S): HELO <identificação SMTP-origem>
SMTP-destino (R): 250 <identificação SMTP-destino>
```

Para o encerramento:

```
S: QUIT
R: 221 ' <identificação SMTP-destino> Service closing transmission
channel
```

Exemplo de uma transação SMTP:

- SMTP-origem (S): MAIL FROM:<Smith.Alpha.ARPA>
- SMTP-destino (R): 250 OK
- S: RCPT TO:Jones@Beta.ARPA
- R: 250 Ok
- S: DATA
- R: 354 Start mail input; end with <CRLF>.<CRLF>
- S: ...etc. etc. etc.
- S: <CRLF>.<CRLF>
- R: 250 Ok



A seguir são apresentados alguns códigos de reply do protocolo SMTP:

- 211 System status, or system help reply
- 220 <domain> Service ready
- 221 <domain> Service closing transmission channel
- 250 Requested mail action okay, completed
- 251 User not local; will forward to <forward-path>
- 354 Start mail input; end with <CRLF>.<CRLF>
- 421 <domain> Service not available, closing transmission channel
- 450 Requested mail action not taken: mailbox unavailable [mailbox busy]
- 451 Requested action aborted: local error in processing
- 452 Requested action not taken: insufficient system storage
- 500 Syntax error, command unrecognized
- 501 Syntax error in parameters or arguments
- 502 Command not implemented
- 503 Bad sequence of commands
- 504 Command parameter not implemented
- 550 Requested action not taken: mailbox unavailable [mailbox not found, no access]
- 551 User not local; please try <forward-path>
- 552 Requested mail action aborted: exceeded storage allocation
- 553 Requested action not taken: mailbox name not allowed [mailbox syntax incorrect]
- 554 Transaction failed

## Protocolo POP3

Permite a uma estação acessar suas mensagens no servidor de modo simples.

Servidor conectado à internet mantém as mensagens.

- Periodicamente buscadas por uma estação.

Não permite muitas opções para a manipulação de mensagens.

- Geralmente as mensagens são baixadas (download) e em seguida removidas do servidor.
- Operações mais complexas são feitas pelo protocolo IMAP4.

O protocolo Post Office Protocol Version 3 (POP3) surgiu para resolver um problema.

Estações que não estão permanentemente conectadas a uma rede IP necessitam receber e tratar suas mensagens. O protocolo POP3 é utilizado por estações para buscar suas mensagens em servidores que as estão armazenando temporariamente.

Em um serviço POP3, existem duas entidades:

- **Cliente:** quem faz uso do serviço POP3;
- **Servidor:** disponibiliza o serviço POP3.

Desse modo, quando um agente de usuário em um cliente deseja enviar uma mensagem a um MTA, uma conexão SMTP é estabelecida e todas as mensagens são entregues a esse MTA, responsável pelo envio.

O funcionamento do protocolo ocorre do seguinte modo:

- O servidor (POP3) em execução escuta a porta TCP 110.
- Quando um cliente deseja utilizar o serviço, uma conexão é estabelecida com o servidor.
- No estabelecimento da conexão, o servidor envia uma saudação.
- Em seguida são trocados pedidos (comandos) e respostas até o fechamento da conexão.

Os comandos utilizados consistem de palavras-chave com três ou quatro caracteres de comprimento, seguidos de um ou mais argumentos, onde cada argumento possui até quarenta caracteres de comprimento.

As respostas consistem de um indicador de status, de uma palavra-chave e de informações adicionais. Todas as respostas terminam com um par de <CRLF>. As mensagens de status são: “+OK” quando positivas e “-ERR” quando negativas. As respostas podem conter múltiplas linhas. Cada linha termina com <CRLF>. A última linha consiste de um código decimal 046, que equivale ao caractere “.” seguido de um par de <CRLF>.

## Comandos POP3

- Autenticação: USER name; PASS string e QUIT.
- Transação: STAT; LIST [msg]; RETR msg; DELE msg; NOOP; RSET e QUIT.

Comandos POP3 (opcionais):

- Autenticação: APOP name digest.
- Transação: TOP msg n e UIDL [msg].
- Respostas POP3:
  - +OK e -ERR.



Os comandos utilizados no protocolo POP3 podem ser divididos em:

Comandos de autenticação (Authorization State) – o cliente se identifica junto ao servidor:

- **USER name:** identifica o usuário.
- **PASS string:** senha do usuário.
- **QUIT:** finaliza a sessão.
- **APOP:** identifica a mailbox e a string MD5 de autorização.

Comandos de transação (Transaction State) – o cliente solicita ações por parte do servidor:

- **STAT:** lista o tamanho das mensagens.
- **RETR msg:** solicita envio de mensagem.
- **RSET:** desmarca mensagem.
- **NOOP:** testa o servidor.
- **TOP msg n:** lista cabeçalho e corpo da mensagem.
- **DELE msg:** marca mensagem para remoção
- **LIST [msg]:** lista mensagens.
- **UIDL [msg]:** informa identificador da mensagem.
- **QUIT:** direciona para Update State.

Comando de atualização (Update State):

- **QUIT:** remove mensagens e finaliza.

Respostas:

- **“+OK”:** resposta positiva a comandos.
- **“-ERR”:** resposta negativa a comandos.

Exemplo:

- Uma sessão pode ter uma autenticação em texto plano ou com a senha criptografada (comando APOP).
- Em seguida, são obtidas informações sobre as mensagens disponíveis (comandos STAT e LIST).
- Na sequência, as mensagens são recuperadas (download) e marcadas para remoção.
- Durante o encerramento as mensagens são excluídas e a sessão encerrada.



Exemplo de uma sessão do POP3:

Servidor (S): <wait for connection on TCP port 110>

Cliente (C): <open connection>

S: +OK POP3 server ready

<1896.697170952@dbc.mtview.ca.us>

C: APOP mrose

c4c9334bac560ecc979e58001b3e22fb

S: +OK mrose's maildrop has 2 messages (320 octets)

C: STAT

```

C: LIST

S: +OK 2 messages (320 octets)

S: 1 120

S: 2 200

S: .

C: RETR 1

S: +OK 120 octets

S: <the POP3 server sends message 1>

S: .

C: DELE 1

S: +OK message 1 deleted

C: RETR 2

S: +OK 200 octets

S: <the POP3 server sends message 2>

S: .

C: DELE 2

S: +OK message 2 deleted

C: QUIT

S: +OK dewey POP3 server signing off

S: +OK 2 320 (maildrop empty)

C: <close connection>

```

## Protocolo IMAP

- Permite a um cliente acessar e manipular mensagens no servidor.
- Atua de maneira a permitir uma manipulação da caixa postal (pastas remotas) como se fossem pastas locais.
- Permite: criar; remover; renomear; verificar por novas mensagens; incluir e remover flags; dar busca em mensagens, cabeçalho, texto e outras partes.
- Mensagens identificadas por números (identificador único).



O protocolo IMAP, assim como o POP3, é responsável pela manipulação de mensagens entre o servidor de e-mail e o cliente. O IMAP é mais recente e possui mais funções que o POP3.

O POP3 foi projetado para manipular mensagens off-line a partir de um único cliente; desse modo, efetua o download da mensagem, que em seguida é removida do servidor.

Já o protocolo IMAP fornece a opção de leitura das mensagens sem necessariamente removê-las do servidor, o que permite, entre outros recursos: a leitura das mensagens a partir de múltiplos clientes instalados, por exemplo, em um computador de escritório e em um computador portátil; utilização de webmail e acesso à conta de correio eletrônico por meio de página web.



Provê ainda uma série de recursos para manipulação de mensagens no servidor, tais como: possibilidade de criar, remover e renomear pastas; ativar ou remover marcadores (flags); Multipurpose Internet Mail Extensions (MIME) parsing: permite a busca e a seleção de mensagens por dados do cabeçalho, corpo ou anexos. Permite ainda suporte a acesso concorrente a caixas de mensagens compartilhadas. O cliente não precisa conhecer o formato utilizado para o armazenamento das mensagens.

Exemplo de uma transação IMAP:

- Uma conexão IMAP consiste em uma saudação inicial seguida por iterações feitas entre um cliente e um servidor.
- Essas iterações possuem uma solicitação enviada pelo cliente e uma resposta enviada pelo servidor.
- Cada resposta inclui os dados solicitados e o resultado da consulta.
- Essas iterações são linhas de texto finalizadas com um <CRLF>.

! O protocolo IMAP4, denominado IMAP versão 4 revisão 1, é orientado à conexão e utiliza a porta TCP 143.

## Conexão e estados do IMAP

1. Conexão sem pré-autenticação: OK.
2. Conexão pré-autenticada: PREAUTH.
3. Conexão rejeitada: BYE.
4. LOGIN ou AUTHENTICATE bem-sucedido.
5. Comando SELECT ou EXAMINE bem-sucedido.
6. Comando CLOSE ou falha no comando SELECT ou EXAMINE
7. Comando LOGOUT ou desligamento do servidor ou conexão encerrada.

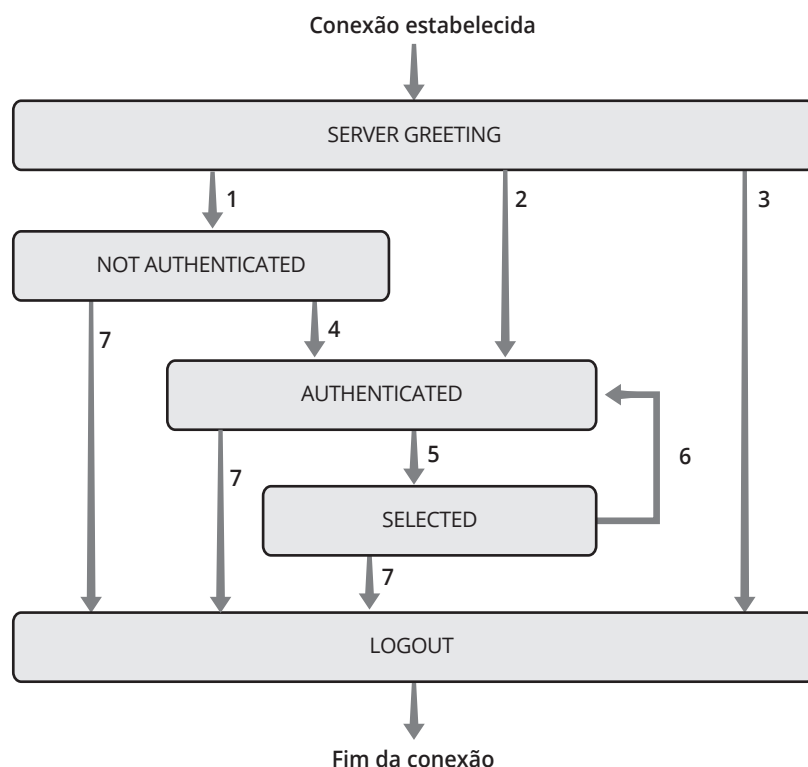


Figura 7.5  
Protocolo IMAP.

Para recuperar uma mensagem ou para efetuar uma entre as várias opções possíveis, é inicialmente estabelecida uma conexão TCP. Após estabelecida uma conexão, ocorre uma operação de saudação (server greeting), e em seguida a conexão muda de estado, podendo estar em um dos seguintes estados:

- **server greeting**: saudação inicial do servidor;
- **not authenticated** (não autenticado): o cliente deve fornecer credenciais (login e senha) antes de encaminhar qualquer comando. Conexões pré-autenticadas passam diretamente ao estado seguinte;
- **authenticated** (autenticado): o cliente deve selecionar uma caixa de mensagens antes de emitir comandos relacionados às mensagens;
- **selected** (selecionado): estado alcançado após uma caixa de mensagens ser selecionada;
- **logout**: estado alcançado após uma operação de LOGOUT do cliente ou por ações unilaterais do cliente ou do servidor.

Comandos válidos em qualquer estado:

- CAPABILITY
- NOOP
- LOGOUT

Estado: Not Authenticated.

- AUTHENTICATE
- LOGIN
- STARTTLS

Comandos:

- **CAPABILITY**: solicita lista de recursos suportados pelo servidor.
- **NOOP**: utilizado para zerar algum contador de inatividade (time-out); não solicita nada.
- **LOGOUT**: o cliente informa ao servidor que terminou suas atividades e deseja encerrar.
- **AUTHENTICATE**: comando para iniciar autenticação (SASL) com o servidor.
- **LOGIN**: utilizado para identificar o cliente, sendo informados usuário e senha em texto plano.
- **STARTTLS**: comando para iniciar autenticação TLS.

Estado Authenticated:

- Comandos válidos em qualquer estado:
- CAPABILITY, NOOP, LOGOUT, SELECT, EXAMINE, CREATE, DELETE, RENAME, SUBSCRIBE, UNSUBSCRIBE, LIST, LSUB, STATUS e APPEND.

Estado Selected:

- Válidos em qualquer estado:
  - CAPABILITY, NOOP e LOGOUT.
- Válidos em Authenticated:
  - SELECT, EXAMINE, CREATE, DELETE, RENAME, SUBSCRIBE, UNSUBSCRIBE, LIST, LSUB, STATUS, APPEND, CHECK, CLOSE, EXPUNGE, SEARCH, FETCH, STORE, COPY e UID.



Comandos:

- ▣ **SELECT**: seleciona uma caixa de mensagens para acessá-las;
- ▣ **EXAMINE**: comando análogo ao SELECT, porém acessa mensagem em modo de leitura (read-only);
- ▣ **CREATE**: cria uma caixa de mensagens;
- ▣ **DELETE**: remove uma caixa de mensagens e todas as mensagens em seu interior. Não remove caixa de entrada (INBOX), somente seu conteúdo;
- ▣ **RENAME**: renomeia uma caixa de mensagens existente;
- ▣ **SUBSCRIBE**: adiciona uma caixa de mensagens a um servidor de notícias;
- ▣ **UNSUBSCRIBE**: remove uma caixa de mensagens de um servidor de notícias;
- ▣ **LIST**: efetua busca baseada em parâmetros, inclusive caractere curinga;
- ▣ **LSUB**: similar ao LIST, porém utiliza os servidores de notícia;
- ▣ **STATUS**: solicita o status de uma determinada caixa de mensagens;
- ▣ **APPEND**: comando para adicionar informação de mensagens a uma determinada caixa de entrada.
- ▣ **CHECK**: solicita uma atualização (checkpoint) da caixa de mensagens atual.
- ▣ **CLOSE**: remove as mensagens marcadas para exclusão.
- ▣ **EXPUNGE**: remove as mensagens marcadas para exclusão, confirmando sequencialmente.
- ▣ **SEARCH**: busca por mensagens de acordo com o critério especificado.
- ▣ **FETCH**: busca dados associados com a mensagem selecionada.
- ▣ **STORE**: armazena na caixa de mensagens as modificações efetuadas nas mensagens.
- ▣ **COPY**: copia uma determinada mensagem para o final da caixa de mensagens de destino.
- ▣ **UID**: funciona de dois modos. No primeiro em conjunto com COPY, FETCH ou STORE, efetua a operação específica sobre a mensagem. No segundo, em conjunto com SEARCH, retorna as mensagens solicitadas.

## Configurações prévias do IMAP

Para funcionar corretamente, o serviço de correio eletrônico depende de configuração adequada de:

- ▣ Hostname; firewall; sincronização de relógio do sistema; Syslog e DNS.
  - ▣ Hostname: servidor deve possuir um FQDN.
  - ▣ Conectividade: firewall deve permitir tráfego na porta 25 SMTP.

Para o funcionamento correto do serviço de correio eletrônico, alguns itens do servidor devem estar adequadamente configurados. Não ajustar esses itens pode causar erros não esperados durante a operação do sistema, como o extravio ou descarte indevido de mensagens pelo MTA de origem, intermediário ou mesmo de destino.

Entre os itens mais importantes estão:

- ▣ Hostname;
- ▣ Conectividade;
- ▣ System Time;
- ▣ Timestamps;



- Syslog;
- DNS.

## Hostname

Um servidor deve possuir um endereço Fully Qualified Domain Name (FQDN), por exemplo *mail.linux.lan*. O hostname é utilizado para saudação a outro servidor. O servidor pode estar repassando mensagens para outros servidores, que verificam o hostname efetuando uma consulta reversa ao DNS, descartando a mensagem cuja origem não foi adequadamente confirmada.

## Conectividade

O firewall deve permitir conexões de entrada e saída na porta 25 utilizada pelo protocolo SMTP.

- System Time e Timestamps: relógio do sistema deve estar sincronizado.
- Syslog: ferramenta para análise e diagnóstico de problemas.
- DNS: configuração afeta desempenho e aceitação de mensagens.



## System Time e Timestamps

É importante que o relógio do sistema esteja sincronizado utilizando, por exemplo, o Network Time Protocol (NTP). A utilização desse tipo de recurso facilita o processo de implantação e ajustes, principalmente ao contatar outros postmasters. Permite ainda uma melhor análise dos cabeçalhos e mensagens de log.

## Syslog

Ferramenta do Sistema Operacional utilizada na análise e diagnóstico de problemas. O servidor Postfix utiliza log padrão do Linux. Logo o serviço syslogd deve estar em execução, e no arquivo */etc/syslog.conf* devem constar entradas associadas ao servidor de mensagens. Por exemplo:

```
# arquivo /etc/syslog.conf
mail.none; authpriv.none; cron.none    -/var/log/messages
mail.*      -/var/log/maillog
cron.*      -/var/log/cron
```

⚠ Neste arquivo deve ser utilizado "<tab>" em vez de espaço.

## DNS

É extremamente importante estar com o serviço DNS do domínio funcionando corretamente antes de ativar o servidor Postfix. Ao enviar mensagens, o MTA precisa resolver o endereço do servidor de destino, assim como o servidor de destino deve efetuar uma confirmação da origem por meio de uma consulta reversa ao DNS.

Servidor de e-mail deve estar registrado no DNS do domínio.

- Permitindo que qualquer servidor na internet possa enviar mensagens a esse servidor.
- Existem três tipos de entradas que devem estar presentes no DNS:





- ▣ A: associa um FQDN a um endereço IP.
- ▣ PTR: associa IP ao FQDN; utilizado para autenticar mensagens.
- ▣ MX: informa o servidor responsável pelas mensagens nesse domínio.



O servidor DNS do domínio deve possuir entradas associadas ao servidor de e-mail do domínio. Isso é necessário porque, na maior parte dos casos, o MTA do destino efetuará uma autenticação da origem verificando a autenticidade do FQDN informado. Logo, o DNS deve ser capaz de informar os servidores responsáveis por mensagens no domínio.

Muitos sistemas não incluem a entrada PTR (utilizada pelo endereçamento reverso), o que acaba causando uma rejeição de mensagens, pois o MTA do destino, ao não obter o FQDN associado, descarta a mensagem supondo que ela seja spam.

Outro cuidado a ser tomado é não incluir entradas CNAME ao servidor de e-mail. O protocolo SMTP exige que o servidor de e-mail seja A ou MX.

Havendo mais de um servidor responsável por mensagens no mesmo domínio, deve ser atribuída prioridade às entradas MX.

Utilize o comando *dig* para testar o funcionamento do servidor DNS.

Exemplo:

- ▣ `dig mail.linux.lan MX`
  - ▣ Retorna informação do servidor de e-mail do domínio linux.lan.
- ▣ `dig -x 200.130.77.75`
  - ▣ Verificação do FQDN associado ao endereço IP.



Utilize sempre o comando *dig* para testar o funcionamento do servidor DNS, verificando as respostas por ele devolvidas.

Exemplo:

```
# dig mail.linux.lan MX
; <<>> DiG 9.2.2 <<>> mail.linux.lan MX
;; global options: printcmd
;; ->>HEADER<<- opcode: QUERY, status:NOERROR, id: 23929
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2
;; QUESTION SECTION
; mail.linux.lan.      IN      MX
;; ANSWER SECTION
; mail.linux.lan. 86400 IN      MX      10      mail.linux.lan.
; mail.linux.lan. 86400 IN      MX      20      mail2.linux.lan.
;; AUTHORITY SECTION
; mar.linux.lan. 86400 IN      NS      mar.linux.lan.
;; Query time: 10 msec
;; Server: 172.25.0.130#53
```

## Postfix

O servidor de e-mail Postfix:

- Consiste em uma Mail Transport Agent (MTA).
- Capaz de aceitar mensagens de outros MTAs e usuários.
- Possui ainda outros subprogramas (distribuição e tratamento de mensagens localmente).
- Consolida-se como principal substituto do Sendmail.
- Propõe velocidade, facilidade de administração e segurança.

A instalação e configuração do Postfix podem ser feitas de maneira intuitiva. A instalação padrão é relativamente segura e simples. Todavia é necessário conhecer os parâmetros para ajustá-los. Recomenda-se que alguns ajustes sejam feitos antes de colocar o servidor em produção. Outros poderão ser feitos com o servidor em produção, especialmente ajustes em filtros de conteúdo e controle de spam.

O Postfix é um Mail Transport Agent (MTA) que envia mensagens de um Mail User Agent (MUA), ou cliente de e-mail, para um servidor remoto utilizando o protocolo SMTP. Pode ainda receber mensagens de outros MTA e encaminhá-las (relay) para outros MTAs. A responsabilidade do Postfix se resume a receber ou enviar mensagens para outros MTAs. A distribuição de mensagens ao usuário final, utilizando o protocolo POP3 ou IMAP, é de responsabilidade de outros servidores.

Existem atualmente muitos MTAs que podem ser utilizados dependendo dos requisitos necessários, incluindo: funcionalidades; facilidade de instalação, configuração e manutenção; robustez; entre outros.

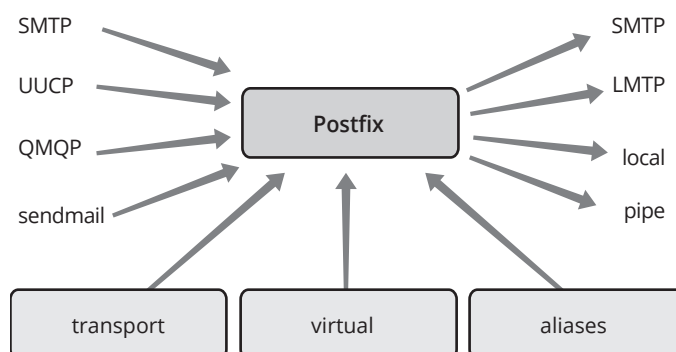
Segurança e estabilidade são outras características positivas do servidor, devido ao seu arranjo modular, e ao fato de que cada componente executa com o mínimo de privilégio no sistema. Possui ainda ampla conectividade com outras aplicações, tais como: banco de dados; serviço de diretório LDAP etc.

O Postfix possui funcionamento semelhante ao de um roteador quando avalia pacotes IP. Ao receber a mensagem, o Postfix avalia o destino (destinatário) e a origem (remetente) para decidir para onde encaminhar a mensagem.

**Figura 7.6**  
Funcionamento do Postfix.

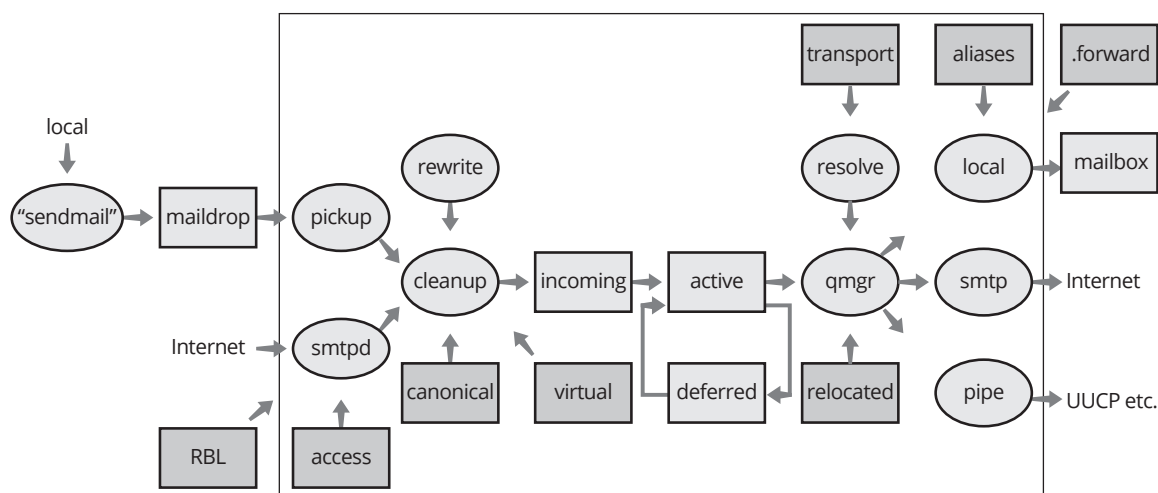


**Figura 7.7**  
Conexões e Mapas.



De modo análogo a um roteador, o Postfix também recebe mensagens por múltiplas origens e pode encaminhá-las também a múltiplos destinos. Uma mensagem pode chegar por meio do sendmail local, por uma conexão SMTP, Unix to Unix Copy (UUCP), ou mesmo Quick Mail Queuing Protocol (QMQP). O destino dessa mensagem pode ser uma caixa local, outro servidor SMTP, uma conexão Local Mail Transfer Protocol (LMTP), ou ainda um pipe para um programa.

Para determinar se as mensagens que chegam serão aceitas, ou mesmo para determinar o método que será utilizado para encaminhar a mensagem, o Postfix faz uso dos “mapas”, que impõem permissões ou restrições baseadas na origem, destino, e mesmo no conteúdo das mensagens.



Fazem parte da arquitetura do Postfix os seguintes componentes:

**Figura 7.8**  
Arquitetura  
do Postfix.

- **sendmail:** comando Postfix para emular o Sendmail. Mantém compatibilidade com programas que chamam por `/usr/sbin/sendmail`. Melhor utilizar SMTP em vez de Sendmail por razão de desempenho;
- **maildrop:** Sendmail envia mensagem para *maildrop queue* e notifica *pickup*;
- **pickup:** coleta a mensagem no *maildrop queue*, efetua o denominado *sanity check* e passa a mensagem ao *cleanup*;
- **smtpd:** cuida da comunicação com outros programas de e-mail; protege o restante do sistema;
- **rewrite:** age sob demanda do *cleanup*, reescrevendo endereços para o padrão `user@fqdn`;
- **cleanup:** verifica formatação da mensagem e a envia para *incoming queue*;
- **incoming:** entrada de mensagem; mensagens não prontas possuem código 0600. Mensagens prontas possuem código 0700 e são coletadas pelo *qmgr*;
- **active:** mensagens prontas para envio, mas não necessariamente em processo de envio;
- **deferred:** mensagens cujo envio falhou para pelo menos um dos destinatários;
- **resolve:** resolve o endereço considerando *transport*, *nexthop*, destinatário e *flags*;
- **qmgr:** parte principal do Postfix, envia tarefas para SMTP, local, LMTP e *pipe*;
- **local:** responsável pela entrega das mensagens, utiliza o formato *mbox* ou *maildir*;
- **smtp:** cliente Postfix para enviar mensagens a destinatários remotos;
- **pipe:** interface de saída para outros mecanismos de transporte de mensagens.

## Instalação do servidor Postfix

Pré-requisitos:

- Instalação do Linux.
- Configurações prévias.
- DNS.





- Remover MTA anterior da instalação do novo MTA.
  - ▣ Não é necessário no Red Hat/Fedora ou Debian.

Instalação:

- Red Hat/Fedora:

```
rpm -Uvh <nome pacote>.rpm
```

- Debian:

```
apt-get install postfix
```

Para a instalação do servidor Postfix, é necessário um servidor com o Sistema Operacional Linux e acesso à internet. Todas as distribuições Linux já vêm com um MTA instalado, que a princípio trata apenas localmente as mensagens. O MTA que normalmente acompanha as distribuições é o Sendmail, Postfix ou Exim. O administrador do sistema pode optar por manter o programa instalado, e nesse caso modifica as configurações que permitem ao servidor executar as tarefas de envio, recebimento ou encaminhamento de mensagens oriundas ou destinadas a outras estações ou domínios.

As distribuições Linux suportam somente um MTA instalado. Desse modo, antes de instalar o novo programa, recomenda-se desinstalar o programa anterior. Para as distribuições Debian e Red Hat/Fedora, não é necessária a etapa de desinstalação. No Debian, ao se instalar o novo programa, o instalador APT automaticamente remove o anterior. Para o Red Hat/Fedora, existe um aplicativo system-switch-mail que permite optar por um entre os MTAs instalados.

Em resumo:

- Red Hat/Fedora: instalar o servidor a partir de um pacote com o comando:

```
rpm -Uvh <nome pacote>.rpm
```

- Debian: instalar o servidor com o comando:

```
apt-get install postfix
```

## Inicialização do servidor Postfix



Para iniciar a execução do servidor Postfix:

- Red Hat/Fedora.
  - ▣ `service postfix start` (*restart* se já estiver iniciado).
- Debian:
  - ▣ `/etc/init.d/postfix start` (*restart* se já estiver iniciado).

Informações são lidas nos arquivos de configuração.

- O principal arquivo é *main.cf*, que contém diretivas (chave = valor) e referências a outros arquivos.

Ao ser iniciado, o servidor Postfix lê as configurações do arquivo de configuração *main.cf*. Opcionalmente, pode-se incluir configurações adicionais em outros arquivos de configuração, referenciando-os em *main.cf*. Para iniciar o servidor, deve-se chamar o binário `/etc/init.d/postfix <parâmetro>`.

O parâmetro pode ser: `start`, `stop`, `restart` ou `reload`. Exemplo: `/etc/init.d/postfix restart`



O arquivo de configuração *main.cf* possui um conjunto de diretivas que norteiam o comportamento do servidor de e-mail Postfix. Nesse arquivo é informado se o servidor cuidará somente de mensagens locais do sistema ou de determinada rede ou domínio; os destinos permitidos para o envio de mensagens; as redes a partir das quais são aceitas mensagens; se esse servidor encaminha mensagens oriundas de outros servidores (relay) etc.

## Configuração do servidor Postfix

Parâmetros a serem ajustados inicialmente.

- **\$myhostname.**
  - ▣ Nome enviado durante a saudação entre MTAs, refere-se ao conteúdo após a @.
  - ▣ Exemplo: usuario@mail.linux.lan
- **\$mydomain** – domínio atual (opcional).
  - ▣ Determinado automaticamente se *myhostname* estiver configurado; conteúdo após o "." depois da @.
- **\$mydestination.**
  - ▣ Aceita mensagens para o domínio listado.
  - ▣ Exemplo: mydestination = \$mydomain



## Configuração do servidor Postfix para um único domínio

O servidor de e-mail Postfix possui centenas de diretivas (parâmetros) que podem ser inseridas em seu arquivo de configuração. Não é necessário configurar cada uma delas, pois todas possuem um valor padrão (default) assumido quando a diretiva é omitida no arquivo de configuração.

Cada diretiva possui o formato: parâmetro = valor. Pode ainda referenciar o conteúdo de outra diretiva: outro\_parametro = \$parametro. Nesse caso, o conteúdo de *parametro* será atribuído a *outro\_parametro*. Pode ainda aceitar múltiplos valores. Exemplos:

```
myhostname = mail.linux.lan
myorigin = $myhostname
mydestination = $myhostname localhost.$mydomain localhost
```

Entre as diretivas que devem ser configuradas inicialmente estão:

- **\$myhostname:** nome enviado durante a saudação entre MTAs, refere-se ao conteúdo após o caractere @. Exemplo: usuario@mail.linux.lan
- **\$mydomain:** domínio atual (opcional), determinado automaticamente se o *myhostname* estiver configurado. Conteúdo após "." depois do caractere @.
- **\$mydestination:** aceita mensagens para o domínio listado. Exemplo: mydestination = \$mydomain.
- **\$myorigin:** domínio que aparece em mensagens enviadas a outras máquinas. Utilizado para garantir que mensagens estejam com remetente no formato FQDN.
- **\$mynetworks\_style:** especifica de onde serão aceitas mensagens. Valores aceitos:
  - ▣ **host:** quando o servidor aceita somente mensagens da máquina local;
  - ▣ **subnet:** quando o servidor aceita mensagens de máquinas pertencentes à sua sub-rede;
  - ▣ **class:** quando o servidor aceita mensagens originárias de servidores com endereço IP de classe igual à sua.

- **\$mynetworks:** lista de clientes autorizados a efetuar relay a partir desse servidor.  
O valor é um endereço de rede no formato Classless Inter-Domain Routing (CIDR), por exemplo: 192.168.0.0/24.

Parâmetros a serem ajustados inicialmente:

- **\$myorigin:**
  - Domínio que aparece em mensagens enviadas a outras máquinas.
  - Utilizado para garantir que mensagens estejam com remetente no formato FQDN.
- **\$mynetworks\_style:**
  - Especifica de onde serão aceitas mensagens.
- **\$mynetworks:**
  - Lista de clientes SMTP autorizados a efetuar relay a partir desse servidor.

Sempre que os arquivos de configuração forem modificados, devem ser lidos novamente para que seus novos valores entrem em vigor. Recomenda-se utilizar nesse caso o parâmetro *reload*, que faz com que o servidor leia as novas informações sem reiniciar. O parâmetro *restart* faz com que o servidor tenha sua execução interrompida e reiniciada, o que causa a indisponibilidade do servidor por alguns momentos.

- É necessário associar nome de e-mail a usuários.
  - Normalmente o Postfix somente entrega mensagens a usuários do sistema.
- Pode ser necessário separar o nome do usuário da conta de e-mail.
  - Restrições administrativas e/ou de segurança.
  - Necessidade de criar arquivo de aliases.
  - Exemplo:
    - usuário uc0003
    - email joao@mail.linux.lan

Uma configuração adicional diz respeito à associação entre contas de e-mail e contas de usuário no sistema. Normalmente o Postfix somente entrega mensagens a usuários do sistema.

Porém, usualmente pode-se desejar desmembrar o nome do usuário da conta no sistema. Existem diversas razões para isso, principalmente questões de segurança e administrativas. A primeira trata da não divulgação das contas de usuário (login) na internet. A segunda pela razão de nem todos os usuários possuírem contas de e-mail.

Para efetuar essa associação, devemos editar um arquivo de mapas denominado */etc/postfix/aliases*, incluindo as associações desejadas.

```
# arquivo /etc/postfix/aliases

usuario1  uc0001
usuario2  uc0002
joao      uc0003
```

Em seguida, deve-se atualizar os mapas com o comando *postalias hash:/etc/postfix/aliases* e, na sequência, recarregar o Postfix.





# Roteiro de Atividades 7

## Atividade 7.1 – Instalação e configuração do Postfix

Para realizar esta atividade, é necessário que o servidor DNS esteja configurado para o domínio *empresa.com.br* com o registro MX *email.empresa.com.br* apontando para o endereço IP do servidor Linux.

O servidor Postfix pode ser instalado de vários modos: a partir do código-fonte ou a partir de instaladores como apt ou synaptic. Nesta atividade, é apresentada a instalação com o apt no Debian, que efetua o download a partir do repositório.

Por questão de segurança, será instalado o suporte ao protocolo TLS e gerado certificado digital para acesso cifrado.

1. Efetue a instalação do servidor Postfix com suporte a TLS.

```
# apt-get install postfix libsasl2-2 sasl2-bin libsasl2-modules libdb-dev procmail
```

Serão realizadas as seguintes perguntas:

- Tipo de configuração? *Site internet.*
- Mail name? *email.empresa.com.br.*

2. Reconfiguração do Postfix:

```
# dpkg-reconfigure postfix
```

Serão realizadas as seguintes perguntas:

- Tipo geral de configuração? *Site internet.*
- Nome de mensagens? *email.empresa.com.br.*
- Para aonde deve ir o mail para o root? *Em branco.*
- Outros destinos para os quais aceitar mensagens? *Em branco (para nenhum) email. empresa.com.br, localhost.empresa.com.br, empresa.com.br, localhost.*
- Forçar atualizações síncronas na fila de mensagem? *No.*
- Redes locais? *127.0.0.0/8, 192.168.1.0/24.*
- Usar o procmail para realizar a entrega local? *Yes.*
- Tamanho máximo da caixa de mensagem: *0.*
- Caractere de extensão de endereço local: *+*
- Qual protocolo de internet utilizar? *Todos.*

3. Modificações no arquivo */etc/postfix/main.cf*:

```
smtpd_banner = $myhostname ESMTPEX $mail_name (Debian/GNU)
biff = no
append_dot_mydomain = no
# paramentos TLS
```





```

smtpd_tls_cert_file = /etc/postfix/ssl/smtpd.crt

smtpd_tls_key_file = /etc/postfix/ssl/smtpd.key

smtpd_use_tls = yes

smtpd_tls_session_cache_database = btree:${queue_directory}/smtpd_
scache

smtp_tls_session_cache_database = btree:${queue_directory}/smtp_
scache

myhostname = email.empresa.com.br

alias_maps = hash:/etc/aliases

alias_database = hash:/etc/aliases

myorigin = /etc/mailname

mydestination = email.empresa.com.br, localhost.empresa.com.br,
empresa.com.br, localhost

mynetworks = 127.0.0.0/8, 192.168.1.0/24

mailbox_size_limit = 0

recipient_delimiter = +

inet_interfaces = all

inet_protocols = all

smtpd_sasl_local_domain =

smtpd_sasl_auth_enable = yes

smtpd_sasl_security_options = noanonymous

broken_sasl_auth_clients = yes

smtpd_recipient_restrictions = permit_sasl_authenticated,permit_
mynetworks,reject_unauth_destination

smtpd_tls_auth_only = no

smtp_use_tls = yes

smtp_tls_note_starttls_offer = yes

smtpd_tls_CAfile = /etc/postfix/ssl/cacert.pem

smtpd_tls_loglevel = 1

smtpd_tls_received_header = yes

smtpd_tls_session_cache_timeout = 3600s

tls_random_source = dev:/dev/urandom

```

4. Execute os seguintes comandos para ativar o suporte a plain login:

```
# echo 'pwcheck_method: saslauthd' >> /etc/postfix/sasl/smtpd.conf
# echo 'mech_list: plain login' >> /etc/postfix/sasl/smtpd.conf
```

5. Adicione o usuário postfix ao grupo sasl:

```
# adduser postfix sasl
```

6. Criação dos certificados digitais:

```
# mkdir /etc/postfix/ssl
# cd /etc/postfix/ssl/

# openssl genrsa -des3 -rand /etc/hosts -out smtpd.key 1024
# chmod 600 smtpd.key

Criando a CA

# openssl req -new -key smtpd.key -out smtpd.csr
# openssl x509 -req -days 3650 -in smtpd.csr -signkey smtpd.key -out
smtpd.crt

# openssl rsa -in smtpd.key -out smtpd.key.unencrypted
# mv -f smtpd.key.unencrypted smtpd.key

Criando o certificado do servidor de email

# openssl req -new -x509 -extensions v3_ca -keyout cakey.pem -out
cacert.pem -days 3650
```

7. Reinicie o servidor Postfix:

```
# /etc/init.d/postfix restart
```

8. Configuração da autenticação com *saslauthd*. Para o Postfix executar na forma *chrooted* em */var/spool/postfix* é preciso realizar as seguintes modificações:

```
# mkdir -p /var/spool/postfix/var/run/saslauthd
```

9. Remova o diretório */var/run/saslauthd*:

```
# rm -rf /var/run/saslauthd
```

10. Crie um link simbólico do diretório *saslauthd*:

```
# ln -s /var/spool/postfix/var/run/saslauthd /var/run/saslauthd
```

11. Edite o arquivo */etc/default/saslauthd* e ative o *saslauthd*.

Configure o "Start" para *Yes* *START=yes* e modifique a linha *OPTIONS="-c"* para *OPTIONS="-c -m /var/spool/postfix/var/run/saslauthd -r"*.

12. Reinicie o *saslauthd*:

```
# /etc/init.d/saslauthd restart
```

13. Para verificar se SMTPd com suporte a TLS está funcionando, execute:

```
# telnet localhost 25
```

14. Após estabelecer a conexão com o Postfix execute o comando *ehlo* para verificar o tipo de servidor e e-mail:

```
ehlo localhost
```

Devem aparecer as linhas 250-STARTTLS e 250-AUTH= LOGIN PLAIN

## Atividade 7.2 – Enviando e recebendo mensagens via Telnet

Nesta atividade será efetuado o teste de envio de mensagem com Telnet, diretamente no servidor de e-mail. Este teste visa averiguar o funcionamento do servidor de e-mail sem a influência de configurações de clientes de e-mail (MUA).

1. Crie os usuários *usuario1* e *usuario2*.



É possível criar o usuário pela linha de comando ou utilizar o *phpldapadmin*. Verifique as atividades do Capítulo 4. Não se esqueça de criar o *home* do usuário e dar as devidas permissões.

2. Conecte-se ao servidor de e-mail estabelecendo uma conexão:

```
$ telnet localhost 25
```

3. Envie um HELO:

```
HELO localhost
```

4. Informe o remetente da mensagem:

```
MAIL From: usuario1@email.empresa.com.br
```

5. Informe o destinatário da mensagem:

```
RCPT To: usuario2@email.empresa.com.br
```

6. Digite o corpo da mensagem:

```
DATA
```

```
Isto eh um teste
```

```
. // digite uma linha somente com ponto "." para encerrar mensagem
```

7. Encerre a conexão:

```
QUIT
```

8. Autentique como *usuario2*:

```
# su - usuario2
```

9. Verifique a chegada de mensagens:

```
# mail
```

### Atividade 7.3 – Enviando e recebendo mensagens

---

Nesta atividade será efetuado o teste de envio de mensagem com as contas de usuários criadas previamente. Esta atividade visa: testar o funcionamento completo da solução de e-mail, incluindo a distribuição de mensagens na caixa do usuário; analisar os logs gerados pelo envio das mensagens; averiguar os aplicativos que estão em execução e a finalidade de cada um.

- Abra um terminal como root e fique monitorando o conteúdo do arquivo `/var/log/mail.log` (pode utilizar o comando `tail`).
- Em outro terminal, logar como `usuario1` e enviar uma mensagem para o usuário2.
- Em um terceiro terminal, logar como `usuario2` e verificar as mensagens.
- O que você consegue identificar analisando os arquivos de log?
- Qual a função de cada um dos aplicativos envolvidos na troca de mensagens (postfix, procmail e bind)?



# 8

## Servidor de correio eletrônico (parte 2)

### objetivos

Administrar o serviço de correio eletrônico, os métodos de entrega de mensagens e o uso de domínios virtuais, controlar o conteúdo, mail gateways e autenticação SMTP, aprender as boas práticas para um servidor de e-mail corporativo e orientações para ajuste de performance, realizar a troca do método de entrega para maildir, configuração dos servidores POP e IMAP e testes com os clientes com esses protocolos.

Configuração do serviço de entrega/maildir; configuração do serviço de recebimento POP/IMAP; configuração do cliente para acesso a e-mail via serviço POP.

### conceitos

### Introdução

O estudo deste capítulo visa complementar o aprendizado da administração do serviço de correio eletrônico, onde são ministrados os métodos de entrega de mensagens e a utilização de domínios virtuais. São abordados o controle de conteúdo, mail gateways e a autenticação SMTP, além de recomendações de boas práticas para um servidor de e-mail corporativo, complementadas com orientações para ajuste de desempenho. Ao final, são praticados a troca do método de entrega para maildir, a configuração dos servidores POP e IMAP e testes com os clientes utilizando esses protocolos.

### Métodos de entrega

- Após receber uma mensagem, um MTA chama outra aplicação para efetuar a entrega na caixa de mensagens do usuário.
- As mensagens destinadas a determinado usuário ficarão em sua caixa postal até que ele as recupere.
- Existem algumas formas de armazenar as mensagens de cada usuário:
  - ▣ mbox
  - ▣ maildir



Após receber uma mensagem, um MTA chama uma aplicação que se encarregará de entregar a mensagem da caixa de mensagens do usuário apropriado. Essas mensagens ficarão disponíveis até que o usuário venha recuperá-las, através dos protocolos POP ou IMAP.

As mensagens deverão então ser armazenadas apropriadamente no servidor, uma vez que poderão ficar disponíveis ao usuário por tempo indeterminado. Existem várias formas de fazer isso. As mais usuais: mbox, maildir ou banco de dados relacional (SQL).

## mbox

- Formato padrão para caixa de mensagens.
- Mensagens de um usuário armazenadas em um único arquivo.
  - ▣ From usuario1@linux.lan Qui Set 12 18:34:00 2006
  - ▣ Received: 12 Set 18:33:59 -0300
  - ▣ From: usuario2@linux.lan
  - ▣ To: usuario1@linux.lan
  - ▣ Subject: olá
  - ▣ Corpo da mensagem



O formato mbox é o formato padrão utilizado em sistemas baseados em Unix. As mensagens de cada usuário ficam em um único arquivo, normalmente localizado em `/var/spool/mail`. Uma variante dessa opção é o uso do formato mbox para armazenamento das mensagens em `$HOME/Mailbox`, a solução mais comum e suportada pela maioria dos MUAs.

O arquivo possui todas as mensagens em sequência. Cada mensagem começa em uma linha iniciada por "From". Essa linha não faz parte do cabeçalho da mensagem utilizada pelo sistema para localizar o início de cada mensagem. Embora o mbox seja amplamente difundido no Linux e funcione adequadamente em um sistema local, não é a melhor forma de tratar mensagens em um servidor de e-mail.



Esse formato não é suportado pelo IMAP. Desse modo, o usuário fica limitado a utilizar o protocolo POP3 para recuperar suas mensagens.

## maildir

Formato recomendado atualmente e suportado pelo protocolo IMAP. A caixa de mensagens do usuário é organizada em várias pastas, geralmente maildir, contendo outras três pastas:

- new
- cur
- tmp



O maildir é o formato recomendado atualmente, por ser o mais confiável. Além disso, é o formato suportado pelo protocolo IMAP. Nesse formato a caixa de mensagens do usuário é organizada em várias pastas. Cada pasta contém as mensagens distribuídas em uma mensagem por arquivo, além de um arquivo de índice. Geralmente uma pasta denominada maildir contém outras três pastas:

- **new**: para mensagens não lidas;
- **cur**: para mensagens lidas;
- **tmp**: para mensagens em processo de entrega.

Um dos maiores benefícios do formato maildir é a confiabilidade, em razão das mensagens estarem distribuídas uma por arquivo, podendo ser localizadas pelo arquivo de índice associado. Além disso, é compatível com o protocolo IMAP, no qual é possível: recuperar somente os cabeçalhos de cada mensagem e manipular cada pasta e as mensagens dentro dela.

A simples instalação do Postfix não cria as pastas utilizadas pelo maildir. Nesse caso existem duas opções: a primeira é usar o utilitário maildirmake, que acompanha o programa Courier Maildrop; a segunda opção é criar as pastas normalmente, o que pode ser feito de vários modos. O mais recomendado é criar as pastas com as devidas permissões em */etc/skel*. Desse modo, cada novo usuário criado terá automaticamente as pastas prontas.

## Servidor para múltiplos domínios

Postfix pode enviar, receber e armazenar mensagens para mais de um domínio.

Para isso, existem dois métodos:

- Virtual alias domains:
  - ▣ Expande o número de domínios pelo qual o servidor responde.
- Virtual mailbox domains:
  - ▣ Cria caixas de mensagens virtuais.
  - ▣ Elimina a necessidade do usuário de possuir conta local na máquina.

O método mais simples de responder por múltiplos domínios é acrescentá-los aos domínios que estão listados no parâmetro de configuração *mydestination* e adicionar os usuários ao arquivo */etc/passwd* do Linux. Porém, se o número de usuários for muito grande, administrá-los é um problema; além disso, não é possível, para o sistema, diferenciar a que domínio pertence um usuário, pois todos serão do domínio principal. Com o uso do virtual alias domain, cada domínio virtual possui suas próprias configurações e os endereços de e-mail são direcionados tanto para contas locais como para endereços remotos. À medida que aumenta o número de domínios e consequentemente o de usuários, torna-se indesejável fornecer uma conta para cada usuário. Com o uso de virtual mailbox domains, além de não ser necessária a tradução de cada endereço de e-mail para um endereço diferente (alias), não é necessária a criação de contas no sistema Linux.

## Domínios virtuais

O Postfix, por padrão, reconhece como destino final somente os nomes especificados no parâmetro *mydestination*; denominado de domínio canônico.

Para domínios adicionais (virtuais), é necessário:

- Estabelecer o alias do nome de domínio virtual.
- Criar um mapa com os destinatários.
- Configurar o Postfix para receber mensagens dos domínios virtuais.

O Postfix reconhece por padrão, como destino final, os nomes especificados no parâmetro *mydestination*, o que é denominado de “domínio canônico”. Para que funcione como destino final para domínios adicionais ou virtuais, é necessário que sejam efetuadas algumas configurações adicionais:

- Estabelecer o alias do nome de domínio virtual;
- Criar um mapa com os destinatários;
- Configurar o Postfix para receber mensagens dos domínios virtuais;
- Recarregar as configurações.



## Estabelecer o alias do nome de domínio virtual

Informar ao Postfix que é o destinatário padrão para outros domínios além do próprio do sistema. No parâmetro *virtual\_alias\_domains* é definida a utilização de um mapa com os domínios virtuais. O valor desse parâmetro é um arquivo contendo os domínios virtuais. Por exemplo:

```
# arquivo /etc/postfix/virtual_alias_domains
                                dominiovirtual.com      20061210
```

Após a criação desse arquivo, deve ser gerado um mapa indexado com o comando *postmap*:

```
# postmap hash:/etc/postfix/virtual_alias_domains
```

## Criar um mapa com os destinatários

Nesta etapa deve ser criado um arquivo */etc/postfix/virtual\_alias\_maps* com os endereços do domínio virtual. Por exemplo:

```
# /etc/postfix/virtual_alias_maps
```

|                               |                      |
|-------------------------------|----------------------|
| postmaster@dominiovirtual.com | usuario1@exemplo.com |
| abuse@dominiovirtual.com      | usuario1@exemplo.com |
| usuario1@dominiovirtual.com   | usuario1@exemplo.com |
| usuario2@dominiovirtual.com   | usuario2@exemplo.com |

Após a criação desse arquivo, deve ser gerado um mapa indexado com o comando *postmap*, conforme segue:

```
# postmap hash:/etc/postfix/virtual_alias_maps
```

## Configurar o Postfix para receber mensagens dos domínios virtuais

Edite o arquivo de configuração *main.cf*, incluindo os parâmetros e valores apropriados:

```
# /etc/postfix/main.cf
virtual_alias_domains = hash:/etc/postfix/virtual_alias_domains
virtual_alias_maps = hash:/etc/postfix/virtual_alias_maps
```

## Recarregar configurações do Postfix

```
# /etc/init.d/postfix reload ou service postfix reload
```

## Testando o envio de mensagem para endereço virtual

```
# echo test | /usr/sbin/sendmail \ postmaster@dominiovirtual.com
```

## Verificando arquivo de log

```
# tail /var/log/mail.log
```

Deve ser visualizado da seguinte maneira:

```
Dez 11 11:20:50 mail postfix/pickup[17850]: B84629AB38: uid=0
from=<root>

Dez 11 11:20:50 mail postfix/cleanup[17863]: B84629AB38: message-
id=<20061210092050. B84629AB38@mail.exemplo.com>
```

```
Dez 11 11:20:50 mail postfix/gmgr[17851]: B84629AB38: from=<root>@mail.exemplo.com>, size 282, nrcpt=1 (queue active)
```

```
Dez 11 11:20:50 mail postfix/local[17866]: B84629AB38: to=<usuario1@exemplo.com>, orig_to=<postmaster@dominiovirtual.com>, relay=local, delay=0, status=sent (mailbox)
```

## Controle de conteúdo



Controle de conteúdo efetuado por ferramentas embutidas ou externas.

- **Embutidas:** resolvem problemas mais simples.
  - ▣ **Restrictions:**
    - ▣ Supervisiona o diálogo SMTP.
  - ▣ **Checks:**
    - ▣ Examina o conteúdo da mensagem e bloqueia mensagens com determinados títulos e tipos de anexos potencialmente perigosos.
- **Externas:** lidam com problemas mais complexos.
  - ▣ Delegam a filtragem de mensagens a aplicações externas.
  - ▣ Onde o controle de cabeçalho e corpo da mensagem já não conseguem atuar.
  - ▣ Em geral efetuam varredura procurando por vírus ou worms e detectando spams.

O assunto de controle de conteúdo em tráfego de mensagens de e-mail é bem amplo. Serão abordados neste curso apenas alguns aspectos que permitem iniciar o controle. O estudo mais aprofundado será necessário à medida que tais controles forem sendo colocados em prática.

No Postfix o controle de conteúdo pode ser dividido em duas partes:

- **Ferramentas embutidas:** visam em geral resolver problemas simples;
- **Ferramentas externas:** lidam com problemas mais complexos.

As ferramentas embutidas efetuam dois tipos de controle:

- **Restrictions:** supervisiona o diálogo SMTP, aceitando ou rejeitando mensagens com base no remetente (mail from), destinatário (rcpt), IP ou hostname.
- **Checks:** examina o conteúdo da mensagem, verificando cabeçalho, corpo da mensagem, anexos e cabeçalhos de mensagens anexadas. Apesar de funcionamento relativamente simples, o administrador deve tomar cuidado com a complexidade das expressões regulares utilizadas. Em geral é utilizado para bloquear mensagens originadas de certos tipos de programas; com determinados títulos (subject); e tipos de anexos potencialmente perigosos.
- As ferramentas externas atuam onde o controle de cabeçalho e corpo da mensagem não conseguem atuar, em geral efetuando varredura procurando por vírus ou worms e detectando spams. Os daemons pipe, SMTP e LMTP podem delegar a filtragem de mensagens a aplicações externas. Em geral instala-se o programa amavisd-new como um intermediário entre os daemons citados e as ferramentas de antispam (por exemplo, SpamAssassin) ou antivírus.

A filtragem ideal não é atingida em um primeiro momento, devendo ser implementada e testada aos poucos:

- Colocar poucas restrições e testá-las.
- Colocar em uso somente após avaliar o impacto.
- Parâmetro *warn\_if\_reject* para testar restrições.



Observação importante: a filtragem ideal não é atingida em um primeiro momento, por isso deve ser implementada e testada aos poucos. Deve-se colocar poucas restrições, testá-las e somente depois colocá-las efetivamente em uso.

É necessário ainda considerar a questão dos falsos positivos e negativos. No primeiro pode-se rejeitar e-mails que a princípio não deveriam estar sendo rejeitados, e no segundo caso ocorre o inverso.

## Mail gateway

Também denominado smarthost.

- Servidor de mensagens que conecta duas redes distintas.
- De um lado, recebe o tráfego de mensagens da internet e, do outro, repassa as mensagens para o servidor de e-mail verdadeiro.
- Protege o servidor tanto de conexões quanto de conteúdos maliciosos.

Um mail gateway também denominado smarthost é um servidor que conecta duas redes distintas. De um lado, recebe o tráfego de mensagens da internet; do outro, repassa as mensagens para o servidor de e-mail verdadeiro. Desse modo, protege o servidor tanto de conexões quanto de conteúdos maliciosos.

Para quem está na rede externa, o mail gateway aparece como o destinatário final das mensagens, já que é ele que consta nos registros **DNS**. Logo, todo o tráfego de mensagens que entra na rede passa por esse servidor.

Por outro lado, para a rede interna o mail gateway aparece como gateway para todas as mensagens que saem da rede. Configurando então o mail gateway para repassar somente mensagens originadas do servidor de e-mail, aliado a um firewall que bloqueie qualquer tráfego SMTP saindo da rede (exceto pelo mail gateway), temos um controle do tráfego SMTP de entrada e saída impedindo o servidor de e-mail de receber diretamente um ataque oriundo da internet. Para isso são necessárias permissões para:

- Servidor interno utilizar mail gateway como relay;
- Estabelecer os domínios que farão relay para a rede interna;
- Estabelecer o host para o qual serão feitos os relays;
- Definir destinatários para os quais será feito relay.

Para seu funcionamento, devemos:

- Configurar servidor interno para utilizar gateway como relay.
- Estabelecer os domínios que farão relay para a rede interna.
- Estabelecer o host para o qual serão feitos os relays.
- Atualizar e gerar um mapa indexado.

Para seu funcionamento, é preciso:

- Definir destinatários para os quais será feito relay.
- Recarregar configurações do Postfix.

1. Configurar servidor interno para utilizar gateway como relay.

Deve-se adicionar no mail gateway a lista de servidores para o qual serão encaminhadas as mensagens que saem. Editar o arquivo *main.cf*, incluindo os endereços permitidos.

```
mynetworks = 127.0.0.0/8 172.25.0.0/24
```



O Postfix possui um parâmetro *warn\_if\_reject* extremamente útil para avaliar o impacto de uma regra de rejeição. Esse parâmetro permite que o log registre a mensagem que seria rejeitada. Desse modo, pode-se avaliar o impacto de determinada regra sem descartar mensagens.



### DNS

(Sistema de Nomes de Domínios, na sigla em inglês) é um sistema de gerenciamento de nomes hierárquico e distribuído com duas definições: examinar e atualizar seu banco de dados e resolver nomes de domínios em endereços de rede (IP).



2. Estabelecer os domínios que serão feitos de relay para a rede interna.

Configurar o mail gateway para aceitar mensagens da rede externa para o servidor interno.

```
relay_domains = exemplo.com
```

3. Estabelecer o host para o qual serão feitos os relays.

Configurar o mail gateway informando para qual servidor na rede interna devem ser encaminhadas as mensagens. Editar o arquivo */etc/postfix/transport*:

```
exemplo.com    smtp:[mail.servidor.exemplo.com]
```

Após a criação desse arquivo, deve ser gerado um mapa indexado com o comando *postmap*:

```
# postmap hash:/etc/postfix/transport
```

Definir o parâmetro *transport\_maps*, no arquivo *main.cf*, ajustando a configuração:

```
transport_maps = hash:/etc/postfix/transport
```

4. Definir destinatários para os quais será feito relay.

Em geral um gateway repassa para o servidor interno tudo o que recebe, inclusive spams, vírus e mensagens para usuários não existentes. Para evitar esse efeito deve ser incluído um mapa de usuários válidos. Criar o arquivo */etc/postfix/relay\_recipients* e inserir os registros:

```
usuariol@exemplo.com    OK
usuario2@exemplo.com    OK
```

Após a criação desse arquivo deve ser gerado um mapa indexado com o comando *postmap*:

```
# postmap hash:/etc/postfix/relay_recipients
```

5. Recarregar configurações do Postfix:

```
# /etc/init.d/postfix reload ou service postfix reload
```

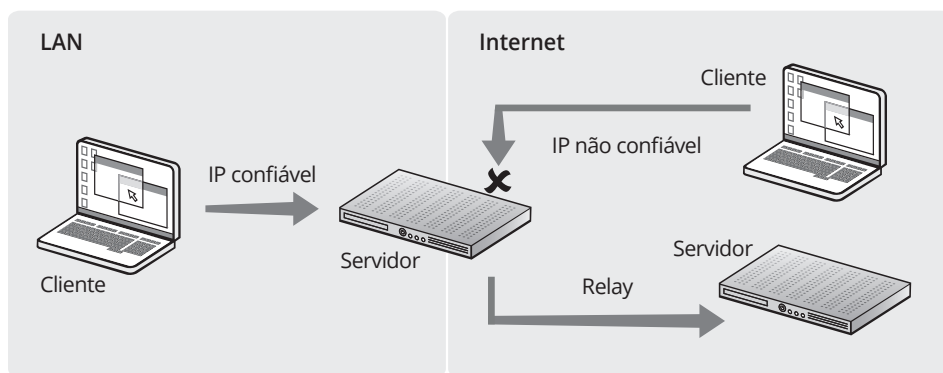
## Autenticação SMTP

- No princípio, os servidores encaminhavam mensagens de qualquer cliente para qualquer destino.
- Com o surgimento do spam, houve a necessidade de se efetuar algum tipo de controle.
- Autenticação SMTP é uma maneira de identificar clientes independente do endereço IP.
- Outras abordagens:
  - ▣ SMTP-after-POP e SMTP-after-IMAP; relay baseado em certificados; e Virtual Private Network (VPN).

A autenticação SMTP (SMTP AUTH) é uma maneira de identificar clientes independentemente de seu endereço IP, tornando possível a um servidor de e-mail efetuar relay de mensagens de clientes, mesmo que estes estejam em redes não confiáveis.

- Identificação de clientes a partir da rede de origem.
- Somente mensagens de determinados segmentos de rede seriam enviadas.





**Figura 8.1**  
Autenticação SMTP.

No SMTP-after-POP e SMTP-after-IMAP, o cliente se autentica no servidor IMAP ou POP. Após a autenticação, o servidor registra em uma base de dados o endereço IP atual do cliente, por um curto período de tempo. Desse modo, quando o cliente tentar enviar uma mensagem, o servidor libera a mensagem após consultar essa base de dados.

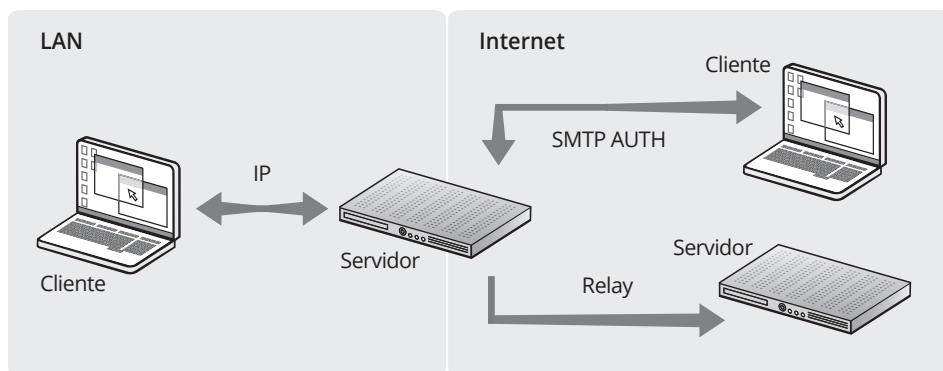
Na autenticação SMTP, o servidor oferece um SMTP AUTH ao cliente, que envia suas credenciais ao servidor. Após a validação, esse certificado permite o encaminhamento de mensagens.

No relay baseado em certificados, o servidor SMTP oferece uma conexão TLS com o cliente, que envia seu certificado para o servidor. Após verificar se o certificado está entre os que podem ser aceitos, passa a encaminhar mensagens desse cliente.

A utilização de VPN concede aos clientes uma conexão ao servidor por meio de rede virtual segura, não tendo relação com configuração SMTP no servidor. Em razão do controle de endereços IP possuídos, pode ser utilizado um controle baseado no endereçamento IP.

SMTP AUTH.

- O servidor oferece SMTP AUTH ao cliente.
  - ▣ Cliente envia suas credenciais ao servidor.
  - ▣ Após validar o certificado, permite o encaminhamento de mensagens.



**Figura 8.2**  
Autenticação SMTP AUTH.

Comparando-se essas quatro opções, podemos chegar às seguintes conclusões:

- **TLS:** funciona bem, no entanto não é suportado por um grande número de clientes, além de o manuseio de certificados ser relativamente complicado;
- **VPN:** exige trabalho adicional de configuração e manutenção para cada usuário;
- **SMTP-after-POP:** não resolve inteiramente o problema, uma vez que é vulnerável a IP spoofing;

- **SMTP AUTH:** resolve o problema inicial proposto. É suportado por uma gama relativamente grande de clientes, entre os quais: Eudora; Fetchmail; JavaMail; Mozilla; Mulbery; Mutt; Netscape; Novell Evolution; Outlook Express; Outlook; PalmOS Eudora; Pine; entre outros.

Apesar do Postfix incluir suporte a SMTP AUTH, essa opção não é habilitada por padrão. É necessário verificar se a biblioteca SASL está instalada, o que pode ser feito com o comando:

```
# ldd `postconf -h daemon_directory`/smtpd
```

A saída gerada deve conter uma linha indicando a presença da biblioteca libssl2.so. Caso ela não esteja presente é necessário reinstalar o Postfix. Instruções podem ser obtidas na documentação disponível.

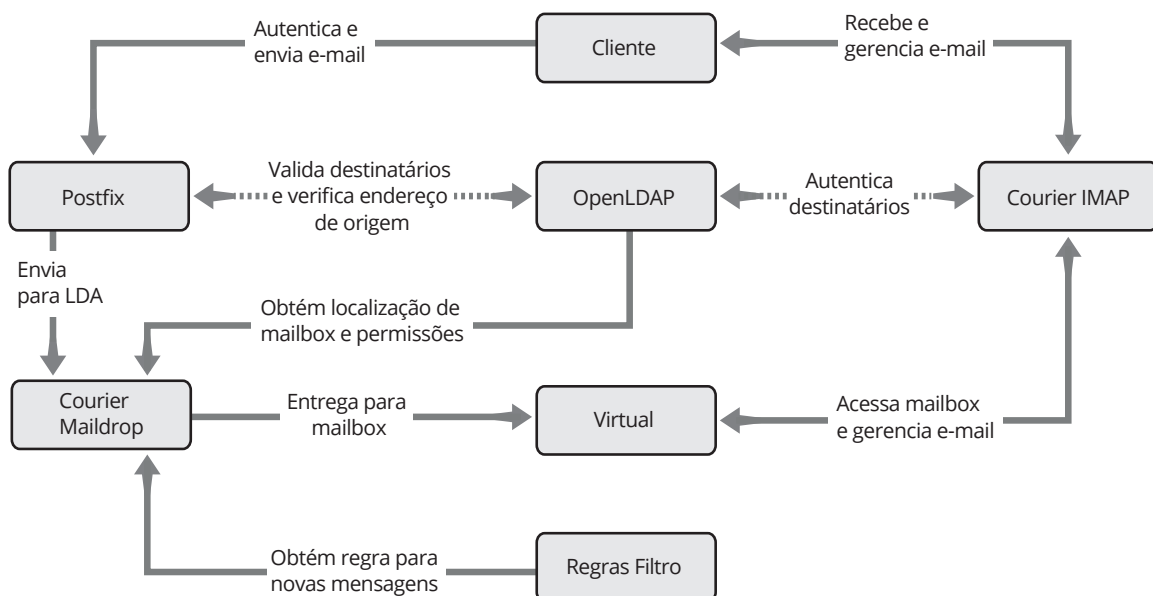
## Servidor de e-mail corporativo

Uma solução completa de e-mail corporativo deve considerar uma série de boas práticas, contendo:

- Postfix.
- Cyrus SASL.
- Courier Maildrop.
- Courier Imap.
- OpenLDAP.

Os itens abordados até permitem a montagem de servidores de correio eletrônico que atendem a uma ampla gama de redes, que vão de algumas dezenas até várias centenas de usuários. Para ambientes em que devem ser suportados milhares de usuários e para fins de facilitar a administração, recomenda-se uma abordagem um pouco diferente. Nessa abordagem é proposta a utilização de algumas ferramentas associadas ao Postfix: Cyrus SASL, Courier Maildrop, Courier Imap e OpenLDAP.

**Figura 8.3**  
Servidor de e-mail corporativo.



- **Postfix:** delega a autenticação do usuário para o servidor LDAP quando os usuários tentam conectar-se a ele utilizando SMTP AUTH. Quando uma mensagem chega, o Postfix consulta ainda o LDAP para verificar os usuários locais e aliases. Ao receber a mensagem, o Postfix entrega-a para o Courier Maildrop, que é o Local Delivery Agent (LDA).
- **Courier Maildrop:** responsável pela entrega local das mensagens, pergunta ao LDAP pela localização das caixas de mensagens. Pode verificar ainda mensagens por meio de regras de filtragem de spam baseadas em uma pasta denominada “.spam”.
- **Courier Imap:** utilizado pelo usuário que se conecta para recuperar suas mensagens. As credenciais do usuário são verificadas junto ao servidor LDAP, que ainda informa onde localizar as mensagens e os identificadores de usuário (UID) e de grupo (GID), que devem ser utilizados.

O propósito de utilização do serviço de diretório OpenLDAP é evitar a necessidade de criação de conta de usuário no sistema para conta de e-mail. Utilizando contas locais baseadas em */etc/passwd*, o Linux está limitado a 65.536 usuários, o que tem ainda implicações de segurança, pois não há associação entre contas de usuário no sistema e contas de e-mail.

## Cyrus SASL

Baseado em três camadas: interface de autenticação; mecanismo e método.

- Interface de autenticação informa ao cliente que uma autenticação e um mecanismo de autenticação estão disponíveis.
- O mecanismo utilizado pelo SMTP AUTH define a estratégia a ser utilizada pela autenticação. Recomenda-se utilizar o PLAIN LOGIN ou CRAM-MD5.
- O método de autenticação manipula o mecanismo cuidando da comunicação entre a aplicação e a base de dados.

Cyrus SASL provê um método de autenticação segura a ser utilizado pelo SMTP AUTH. É baseado em três camadas: interface de autenticação; mecanismo e método.

O propósito da interface de autenticação é informar ao cliente que estão disponíveis uma autenticação e um mecanismo de autenticação. Essa interface é implementada pelo serviço e protocolo em uso, como por exemplo o SMTP.

O mecanismo utilizado pelo SMTP AUTH define a estratégia a ser utilizada pela autenticação. Recomenda-se utilizar o PLAIN LOGIN ou CRAM-MD5, embora outros métodos possam ser suportados, tais como: GSSAPI, KERBEROS\_V4, NTLM e OTP.

No PLAIN LOGIN as credenciais são enviadas utilizando PLAIN como mecanismo, através de strings com codificação de 64 bits que podem ser facilmente decodificadas e utilizadas inclusive por clientes que não seguem a RFC, como por exemplo o Outlook Express.

O CRAM-MD5 e seu sucessor DIGEST-MD5 utilizam um mecanismo de senha (segredo) compartilhada baseada em desafio. O servidor gera um desafio baseado em um segredo e o cliente responde provando que conhece o segredo.

O método de autenticação manipula o mecanismo, cuidando da comunicação entre a aplicação que oferece a autenticação SMTP e a base de dados que armazena as credenciais. São dois os métodos: *saslauthd* – Daemon que executa com privilégios de root e pode acessar alguns tipos de bases, como por exemplo */etc/shadow*; e *auxprop* – serviço para trabalhar em conjunto com plugins. Cyrus SASL pode utilizar:

- **Imap:** servidor Imap para verificação de credenciais;



- **Kerberos:** verifica tickets Kerberos;
- **Ldap:** verifica credenciais no LDAP;
- **Pam:** utiliza Pluggable Authentication Modules (PAM);
- **Passwd/shadow:** base de dados do sistema;
- **Sasldb2:** base de dados própria do Cyrus SASL;
- **SQL:** dados armazenados em servidores SQL.

## Courier Maildrop

- Agente local para a entrega de mensagens.
- Recebe mensagens do Postfix e as armazena em caixas de mensagens no formato maildir.
- Pode ainda aplicar filtros nas mensagens.



Courier Maildrop é um agente local para a entrega de mensagens que recebe mensagens de um agente de transporte (Postfix) e armazena-as em caixas de mensagens no formato maildir. Pode ainda aplicar filtros nas mensagens.

Um recurso adicional interessante é a utilização conjunta com cotas em diretórios, uma vez que nesse formato as mensagens de cada usuário ficam em diretórios separados.

A instalação do Courier Maildrop exige a criação de usuário e grupo específicos:

```
# useradd -u 1003 vmail
# groupadd -g 1003 vmail
```

Para efetuar a instalação, utilize as seguintes opções quando for construí-lo:

```
# ./configure --enable-restrict-trusted=1 --enable-trusted-
users='root vmail' --enable-trusted-groups='root vmail' --enable-
maildirquota --enable-trashquota --enable-maildropldap
```

Após essa etapa, execute os comandos *make install-strip* e *install-man* para instalar os binários e o manual. Efetue ainda os seguintes ajustes de permissão em razão do daemon pipe do Postfix (que não executa como root):

```
# chmod 750 /usr/local/bin/maildrop
# chmod u+s /usr/local/bin/maildrop
# chown root:vmail /usr/local/bin/maildrop
```

## Courier Imap

- Oferece os serviços:
  - POP.
  - POP-SSL.
  - IMAP.
  - IMAP-SSL.
- Suporta o formato maildir.



Courier Imap suporta o formato maildir e oferece os serviços POP, POP-SSL, IMAP e IMAP-SSL para os clientes que a ele se conectam para manipulação das mensagens.





Para a instalação, utilize as seguintes opções quando for construí-lo:

```
# ./configure --enable-workarounds-for-imap-client-bugs --enable-unicode --without-authpgsql --without-socks --with-redhat make
```

O parâmetro `--with-redhat` deve ser utilizado somente em distribuições baseadas em Red Hat. Em seguida, execute os comandos *make install* e *install-configure*.

É necessária ainda a criação dos certificados para autenticação, que pode ser feita manualmente ou por meio do utilitário *mkimapdcert*, executado automaticamente ao iniciar o *imapd-ssl*.

## Recomendações de tuning

- Ajustes básicos.
- Identificação de gargalos.
- Ajustes para alto volume de tráfego.



O Postfix é rápido, e essa é uma de suas principais características. Todavia, como muitos outros programas, pode ter seu desempenho aumentado com alguns ajustes. Além disso, algumas situações, como limitações de hardware, software ou flutuações bruscas (como um pico no volume de spam) podem afetar sua performance.

Entre os ajustes básicos, alguns itens sempre merecem ser verificados para maior garantia de ausência de problemas inesperados. Podemos citar: ajustes em consultas DNS; verificar se o servidor está atuando como relay, recusando mensagens para usuários inexistentes, bloqueando mensagens oriundas de redes da lista negra ou reduzindo a frequência de retransmissões.

Gargalos afetam o desempenho do sistema, eventualmente sobrecarregando o servidor e causando nos usuários a percepção da ocorrência de problemas. Ao chegar ao servidor, durante seu processamento, uma mensagem é movida de uma fila para outra. As filas são:

- **Incoming:** todas as mensagens que entram são colocadas nessa fila pelo daemon *clean*, e quando ficam prontas o *qmgr* é notificado;
- **Deferred:** mensagens que já foram enviadas a alguns destinatários, mas com falha para um ou mais destinatários, aguardam nessa fila por nova varredura do *qmgr*;
- **active:** mensagens nessa fila estão prontas para envio, mas ainda não foram enviadas;
- **maildrop:** aguardam nessa fila mensagens enviadas pelo comando *sendmail* e que ainda não foram enviadas para as filas do Postfix pelo daemon *pickup*.

Para alto volume de tráfego, considere utilizar um servidor em separado para mensagens que chegam. Separe ainda, se for o caso, a máquina que efetua filtragem de vírus, desabilitando nessa máquina consultas DNS, uma vez que ela encaminhará as mensagens sempre para um mesmo destino.

## Ajustes em consultas DNS

- DNS deve estar funcionando adequadamente.
  - Efetuar teste com o comando *dig*.
- Verificar:
  - Arquivo *resolv.conf*.
  - Saturação de tráfego na rede.



- ▣ Configurações do firewall.
- ▣ Disponibilização de um servidor cache DNS.



Quando o sistema possui alto volume de tráfego, é importante que o DNS esteja configurado adequadamente, sobretudo se a verificação de origem através do Fully Qualified Domain Name (FQDN) está sendo efetuada.

Utilize o comando *dig* para efetuar uma consulta DNS. Recomenda-se efetuar esse teste duas vezes verificando o tempo de resposta na primeira e segunda consultas. Os valores obtidos variam de rede para rede; no entanto, no primeiro teste o valor deverá ser superior a 100 ms, enquanto que em uma segunda consulta deverá ficar por volta de 10 ou 20 ms, indicando que o servidor DNS está efetuando cache corretamente.

Persistindo os problemas, verifique:

- ▣ As configurações do arquivo *resolv.conf*, principalmente os servidores indicados na diretiva *nameservers*;
- ▣ Problemas de tráfego na rede; caso a rede esteja saturada, verifique a possibilidade de otimizar a utilização da banda, ou mesmo priorizar o tráfego dos servidores DNS;
- ▣ Configurações do firewall;
- ▣ Disponibilização de um servidor cache DNS caso o servidor DNS esteja sobrecarregado ou não esteja no mesmo segmento de rede do servidor de e-mail e se queira aliviar o tráfego na rede e melhorar o tempo de resposta.

Verificar se o servidor está atuando como relay.

- ▣ Carga desnecessária no servidor e na rede.
- ▣ Servidor acabará incluído em listas negras.
- ▣ Permita o envio de mensagens somente de clientes:
  - ▣ Listados no parâmetro *mynetworks*.
  - ▣ Autenticados via SMTP.
  - ▣ Autenticados com certificados TLS.



Verifique se o servidor não está atuando como um relay aberto. Deixar o servidor efetuando relay de qualquer origem é extremamente prejudicial, pois além de submeter o servidor de e-mail a uma carga desnecessária, essa atitude fará com que o servidor acabe sendo incluído em listas negras (black lists), fazendo com que as mensagens legítimas da rede sejam recusadas por muitos destinos.

Permita somente o envio de mensagens de clientes listados no parâmetro *mynetworks* no arquivo de configuração; de clientes autenticados via SMTP e de clientes que se autenticarem utilizando certificados TLS.

Recusando mensagens para usuários inexistentes.

- ▣ Servidor deve recusar mensagens para usuários inexistentes.
  - ▣ Caso aceite, enviará mensagens com notificações de não entrega de mensagens, causando acúmulo de mensagens na fila de saída.
- ▣ Configure os parâmetros:
  - ▣ *local\_recipient\_maps*
  - ▣ *relay\_recipient\_maps*



Configure o servidor para recusar mensagens de usuários inexistentes. Caso o Postfix aceite esse tipo de mensagem, o sistema acabará enviando mensagens com notificações de não entrega de mensagens (undeliverable messages), causando acúmulo de mensagens na fila de saída. Além disso, essas mensagens ocuparão grande espaço no sistema.

Outra questão surge quando o servidor está atuando como relay; nesse caso, o servidor de destino acabará enviando de volta mensagens (bounce) não entregues, utilizando o Return-Path do cabeçalho. Para esses ajustes, configure os parâmetros *local\_recipient\_maps* e *relay\_recipient\_maps* (caso o servidor esteja atuando como relay para outro servidor na rede interna).

Caso o servidor esteja recebendo muitas mensagens de volta (bounce), e isso esteja se tornando um problema, verifique na internet a lista de servidores que não aceitam suas mensagens de volta. Caso o servidor esteja listado nessa lista, inclua-o em sua lista negra, o que pode ser feito automaticamente com o parâmetro:

```
check_rhsbl_sender dsn.rfc-ignorant.org
```

Bloqueando mensagens oriundas de redes da lista negra.

- Existem atualmente vários tipos de listas negras, inclusive de DNS.
  - ▣ A utilização de listas reduz a carga de spam no servidor.

Reduzindo a frequência de retransmissões.

- Quando há grande número de mensagens não entregues na primeira tentativa:
  - ▣ Usar um relay para fallback.
  - ▣ Aumentar o tempo entre as tentativas.

Existem atualmente vários tipos de listas negras, inclusive de DNS, que listam endereços IP, faixas de endereços IP ou mesmo domínios.

Essas listas contêm poucos falsos positivos em razão dos critérios utilizados para inclusão. A utilização dessas listas fará com que o servidor receba carga menor de spam.

Outra questão surge quando o servidor tem grande número de mensagens que não são entregues na primeira tentativa. Existem duas alternativas:

- Usar um relay para *fallback*; delega tentativas malsucedidas a outro servidor, que se encarrega desse tipo de mensagem.
- Aumentar o tempo entre as tentativas pelo parâmetro *maximal\_backoff\_time* para diminuir a frequência com que uma mesma mensagem entra novamente na fila de saída. Faz com que tentativas de envio de mensagens a servidores que estejam fora do ar sejam feitas em intervalos maiores.

## Gargalos na incoming queue

Lentidão em consultas externas. Exemplo: LDAP.

As mensagens que chegam são colocadas nessa fila pelo *cleanup* com permissão 0600 até que estejam completas e prontas para processamento, quando sua permissão é alterada para 0700. Em condições normais a fila *incoming* quase não possui mensagens 0600. Caso ocorram picos de recebimento de mensagens anterior da capacidade do qmgr, essa fila pode crescer consideravelmente. Nesse caso, o que pode estar retardando o qmgr é o serviço *trivial-rewrite*, sobretudo por lentidão nas consultas LDAP ou SQL.

## Gargalos na maildrop queue

Retardo decorrente de excesso de mensagens ou de consumo excessivo de CPU devido às verificações do *pickup*.



Mensagens enviadas pelo comando *sendmail* e ainda não coletadas pelo daemon *pickup* permanecem na fila *maildrop*. Mensagens enviadas pelo comando *sendmail* mesmo com o Postfix fora de execução vão para essa fila. O *pickup* periodicamente verifica essa fila ou realiza a verificação quando é notificado da chegada de novas mensagens pelo *postdrop*. Executando em uma única thread, processa somente uma mensagem por vez, verificando cabeçalho, corpo e outras informações, demandando muito processamento de entrada e saída de disco. Retardos causados nessa fila são decorrentes de excesso de mensagens ou consumo excessivo de CPU devido às verificações do *pickup*.

Convém observar que quando a fila *active* estiver cheia, o *cleanup* diminui a injeção de mensagens. A fila *active* possui limite de 20 mil mensagens.



Evite que uma quantidade excessiva de mensagens seja entregue via *pickup*. Caso necessário, utilize opções alternativas de injeção de mensagens por meio de programas como *mini\_sendmail*.

## Gargalos na deferred queue

Havendo grande volume de mensagens nessa fila, pode-se tentar ajustar diminuindo o tempo mínimo e aumentando o tempo máximo das mensagens, agilizando o envio de mensagens com pouco tempo na fila, sem provocar muitas tentativas para todas as mensagens.



Quando o Postfix não consegue entregar a mensagem para algum dos destinatários, esta é colocada na fila *deferred*. O *qmgr* verifica periodicamente essa fila conforme especificado no parâmetro *queue\_run\_delay*. Uma fração das mensagens é periodicamente reinjetada na fila *active* de acordo com o tempo de espera, que varia entre os valores de *minimal\_backoff\_time* e *maximal\_backoff\_time*, respectivamente o tempo mínimo e máximo que cada mensagem aguarda antes de ser recolocada para envio. Cada nova tentativa dobra o tempo de vida da mensagem na fila. Mensagens com tempo menor possuem tentativas mais frequentes e mensagens mais antigas têm tempo maior entre as tentativas. Havendo grande volume de mensagens nessa fila pode ser feita a tentativa de ajuste diminuindo o tempo mínimo e aumentando o tempo máximo das mensagens, o que agiliza um pouco o envio de mensagens com pouco tempo na fila sem provocar muitas tentativas para todas as mensagens.

## Gargalos na active queue

- Quando o destino funciona a uma velocidade muito menor, causa acúmulo nessa fila.
- Caso a fila *active* esteja realmente lenta:
  - ▣ Reduzir a injeção de mensagens nessa fila.
  - ▣ Aumentar o *throughput* incrementando a concorrência (quantidade de processos SMTP em execução) ou reduzindo a latência (melhorias no DNS, mapas).



O *qmgr* tenta enviar cada mensagem dessa fila para seu destino. Quando o destino funciona com velocidade muito menor, provoca certo acúmulo nessa fila. Caso o destino saia do ar por alguns momentos, o *qmgr* considera esse destino como morto e transfere todas

as mensagens para *deferred*, o que libera a fila *active*, mas entrega muitas mensagens na fila *deferred*. Caso a fila *active* esteja realmente lenta, existem dois modos de resolver o problema: reduzir a injeção de mensagens nessa fila ou aumentar o *throughput*. Para aumentar o *throughput* pode ser incrementada a concorrência (quantidade de processos SMTP em execução) ou reduzida a latência (melhorias no DNS, mapas).

O aumento da concorrência pode ser feito pela modificação do valor no parâmetro *default\_process\_limit* no *main.cf*.

Para modificação baseada no destinatário, utilize o utilitário *qshape*, que apresenta na forma de uma tabela o total de mensagens na fila baseadas no destinatário. Podem ser utilizados os comandos:

```
# qshape -s hold
# qshape deferred
# qshape incoming active deferred
```

Por último, deve-se evitar recarregar ou reiniciar o servidor, quando possível. Apesar de a fila *active* em memória estar vazia, a fila *active* em disco pode conter muitas mensagens, que são devolvidas à fila *incoming*, demandando muito consumo de memória.



# Roteiro de Atividades 8

## Atividade 8.1 – Configuração da modalidade de entrega maildir

---

### Pré-condição

Para realizar esta atividade é necessário que o servidor DNS esteja configurado para o domínio empresa.com.br com o registro MX email.empresa.com.br apontando para o endereço IP do servidor Linux.

Ao instalar e configurar o serviço de correio eletrônico, a modalidade de entrega das mensagens é mbox.

1. Crie a estrutura de diretórios do maildir no home dos usuários usuario1 e usuario2.
2. Configure o postfix para entregar as mensagens na modalidade maildir.
3. Envie uma mensagem de teste e verifique se o a entrega foi realizada.

## Atividade 8.2 – Configurando servidores POP e IMAP

---

Os protocolos POP e IMAP são utilizados pelos programas cliente (MUA) para recuperar mensagens armazenadas no servidor de e-mail. Nesta atividade são configurados os servidores POP e IMAP, que são testados com Telnet.

1. Instale o Courier-IMAP/Courier-POP3.

Serão realizadas as seguintes perguntas:

- Criar os diretórios para administração web? Responda “No”.
- Requer certificado SSL? Responda “Ok”.

2. Para acessar o servidor POP3, digite o comando a seguir. Deverá retornar uma mensagem informando que o servidor está pronto:

```
# telnet localhost 110 // 110 porta utilizada pelo protocolo POP3
Connected do localhost
Escape character is '^]'
+OK Hello there
```

3. Para acessar o servidor IMAP, digite o comando a seguir. Deverá retornar uma mensagem informando que o servidor está pronto:

```
# telnet localhost 143 // 143 porta utilizada pelo protocolo IMAP
Connected do localhost
Escape character is '^]'
+ OK [CAPABILITY IMAP4rev1 UIDPLUS CHILDREN NAMESPACE .....
```



#### 4. Teste de autenticação do POP3:

```
# telnet localhost 110
Trying 0.0.0.0...
Connected to 0.
Escape character is '^]'.
+OK Hello there
user usuariol
+OK Password required.
pass senha
+OK logged in.
list
+OK POP3 clients that break here,
they violate STD53.
1 401
```

### Atividade 8.3 – Configurando a autenticação POP e IMAP no LDAP

Altere a configuração do SASL para que possa realizar a autenticação a partir da base do LDAP.

1. Instale o pacote *libsasl2-modules-ldap* para permitir que o SASL realize consultas à base do LDAP.

```
# apt-get install libsasl2-modules-ldap
```

2. Altere o arquivo */etc/defaults/saslauthd* e atribua o valor *ldap* para o parâmetro MECHANISMS.

```
# vi /etc/defaults/saslauthd
MECHANISMS="ldap"
```

3. Adicione as seguintes linhas no arquivo */etc/postfix/sasl/smtpd.conf*:

```
# vi /etc/postfix/sasl/smtpd.conf
pwcheck_method: saslauthd
mech_list: plain login
auxprop_plugin: ldap
ldapdb_uri: ldap://192.168.1.1
ldapdb_id: admin
ldapdb_pw: rnpesr
ldapdb_mech: DIGEST-MD5
```

4. Crie o arquivo */etc/saslauthd.conf* e coloque as seguintes informações nele:

```
# vi /etc/saslauthd.conf

ldap_servers: ldap://192.168.1.1/

ldap_search_base: ou=people,dc=empresa,dc=com,dc=br

ldap_auth_method: bind

ldap_filter: uid=%U
```

5. Reinicie o *saslauthd* e o Postfix:

```
#!/etc/init.d/saslauthd restart

#!/etc/init.d/postfix restart
```

6. Testando o funcionamento:

```
# testsaslauthd -u aluno -p rnpesr

0: OK "Success."
```

## Atividade 8.4 – Utilização de clientes POP e IMAP

---

Os programas cliente (MUA) utilizam-se dos protocolos POP ou IMAP para recuperar mensagens no servidor de e-mail. Nesta atividade é configurado um cliente para o recebimento de mensagens com esses protocolos.

1. Instale um cliente de e-mail como o Icedove (Thunderbird) ou Evolution na máquina cliente Linux;
2. Configure o cliente de e-mail para enviar e receber as mensagens do usuário2.
3. Instale o cliente de e-mail Thunderbird na máquina Windows;
4. Configure o cliente de e-mail para enviar e receber mensagens do usuário1;
5. Troque mensagens entre os usuários e verifique o comportamento do servidor.





# 9

## Servidor proxy Squid

### objetivos

Entender o que é servidor proxy, fazer a instalação e configuração do servidor Squid na versão compilada, configurar regras de acesso ACL, emitir relatório de acesso à internet usando SARG e ativar o serviço de proxy transparente.

### conceitos

Proxy, solução Squid, configurações no servidor e no cliente, restrição de acesso a conteúdos, redirecionamento, proxy transparente.

### Introdução

Conceitos iniciais:

- Proxy.
- Proxy Squid.

Configuração Squid:

- Servidor e cliente.
  - Restringindo o acesso à internet.
  - Restringindo o acesso a determinadas páginas.
  - Proxy transparente.

Este capítulo aborda os conceitos associados ao serviço proxy e sua aplicabilidade, sendo estudados:

- Conceitos iniciais de proxy;
- Solução Squid;
- Configurações necessárias no servidor e no cliente;
- Restrição de acesso aos conteúdos, incluindo redirecionamento;
- Proxy transparente.

Em seguida são praticados:

- Instalação e configuração do Squid;
- Configuração do navegador;
- Utilização de proxy transparente e de listas de controle de acesso que limitam e modelam o tráfego, considerando vários aspectos.



## Proxy

- Tem por função limitar o tipo de tráfego.
- Atua na camada de aplicação (7), analisando o conteúdo dos pacotes.
  - Pode barrar pacotes que contenham, por exemplo, palavras inapropriadas.
  - Diferentemente de um filtro de conteúdo que analisa tráfego baseado em camada de rede (3) ou de transporte (4).

O serviço proxy tem por função limitar o tipo de tráfego que passa por ele. Instalado na borda de uma rede, efetua o monitoramento dos pacotes e, se for o caso, barra o trânsito.

De modo análogo ao filtro de pacotes que, baseado na faixa de endereços IP (camada 3) ou porta (camada 4), impede o tráfego de determinadas informações, o proxy atua em camadas mais altas, podendo limitar determinados tipos de protocolos, por exemplo o ICMP (ping). Por funcionar analisando o tráfego, pode examinar o conteúdo do pacote na camada 7 (aplicação). Um exemplo clássico é procurar nos pacotes por palavras que constem em uma lista proibitiva, tal como “sexo”. Todo pacote que contiver essa palavra será descartado, impedindo o acesso a páginas que contenham conteúdo impróprio ou estranho às necessidades da rede, seja uma rede residencial, de empresa ou de escola.

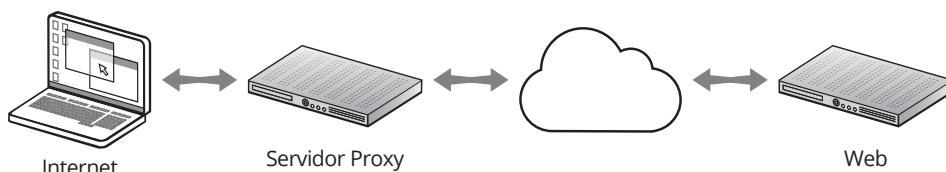
Web proxy cache:

- Atua como cache.
- Reserva uma área em memória para armazenar os conteúdos acessados com maior frequência.
- Ao buscar uma informação que tenha sido acessada previamente, o servidor proxy cache a entrega diretamente sem buscá-la na internet.

Outra finalidade do proxy é atuar como cache. Nesse caso, o servidor reserva uma área em memória para armazenar os conteúdos estáticos acessados com maior frequência pelos usuários de rede interna. Quando o usuário busca por determinada informação, o servidor proxy cache o entrega diretamente sem acessá-lo na internet. Considere por exemplo um grande portal de notícias da internet. A primeira pessoa a acessá-lo fará com que o conteúdo dessa página fique armazenado no cache do servidor. As próximas pessoas que acessarem essa mesma página, dentro do tempo de expiração programado, obterão o conteúdo do servidor, em vez do conteúdo da internet.

Portanto, essas duas soluções apresentam, por motivos diferentes, melhoria no tráfego da rede. O proxy bloqueia o tráfego considerado inadequado pela política de utilização da rede da empresa, enquanto o cache contribui para reduzir o montante de tráfego no link externo da rede.

- Acesso à internet efetuado através de um proxy.
- Geralmente o proxy está associado a um firewall.
- Funciona como filtro de conteúdo e como web proxy cache.



**Figura 9.1**  
Funcionamento do proxy cache.

A utilização de um web proxy cache possibilita grande economia de recursos, com impacto tanto na velocidade quanto no controle de acesso.

Na velocidade, o impacto ocorre de duas maneiras. Na primeira, os usuários conseguem acessar as páginas mais rápido, uma vez que elas são carregadas a partir do cache, que está mais próximo do usuário. A segunda maneira é indireta e se aplica a todos os usuários que acessam dados na internet, que por estar com carga menor do que estaria sem o cache apresenta desempenho mais elevado.

O controle de acesso é cada vez mais necessário, em empresas de qualquer tamanho. Com a maior penetração obtida pela internet, os benefícios proporcionados são evidentes, como:

- Agilidade na troca de informações com outras empresas, funcionários e clientes;
- Relacionamentos pessoais e familiares;
- Comércio eletrônico;
- Distribuição e compartilhamento de conteúdo etc.

Aliado a esses fatores, há a ampliação constante da largura de banda por parte de empresas e usuários domésticos, incentivando-os a utilizar serviços antes inviáveis. Esse cenário gera demanda sempre crescente de largura de banda.

Desse modo, é extremamente necessário, em especial por parte de empresas, a definição de uma política de utilização de rede em que todos os envolvidos tenham plena consciência do modo como devem utilizar os recursos disponibilizados. Faz-se necessário o controle do registro de todos os acessos, bloqueando aqueles considerados indevidos, reduzindo a utilização de largura de banda em ações estranhas às atividades da empresa, mantendo-a disponível e melhor preparada para a prática de atividades legítimas. Essa política também diminui a circulação de vírus, worms, programas piratas e outros males que apresentam riscos às empresas.

## Proxy Squid

O Squid é um dos proxies mais utilizados na internet.

- Considerado simples e confiável.
- Praticamente obrigatório em qualquer tipo de organização com serviços de internet.
  - De pequenas empresas aos grandes provedores de acesso.
- Funciona tanto como um servidor proxy quanto como um web cache.
- Vantagens de um proxy (entre eles o Squid):
  - Capacidade de armazenar documentos da internet.
  - Bloquear o acesso a determinadas páginas.

Squid é um dos proxies mais utilizados na internet. Considerado simples e confiável, é um recurso praticamente obrigatório em qualquer tipo de organização que utilize serviços de internet, desde pequenas empresas aos grandes provedores de acesso.

Foi originado de um projeto da ARPA, cujo mentor foi Duane Wessels, do National Laboratory for Applied Network Research, tendo posteriormente obtido a denominação de Squid. É tanto um servidor proxy quanto um web cache. Como proxy possui características especiais para a filtragem de pacotes, suportando vários protocolos, como HTTP e FTP. Pode ainda atuar como um proxy reverso, funcionando nesse caso como um acelerador para um servidor web.

A grande vantagem de um proxy (como o Squid) é a capacidade de armazenar documentos da internet. Possui também o recurso de criação de regras de acesso, que permitem ou bloqueiam o acesso a determinadas páginas. Com isso, pode-se vetar a navegação em sites



pornográficos, salas de bate-papo, serviços de mensagens instantâneas ou de compartilhamento de arquivos. Frequentemente é associado a um firewall, estando inclusive instalado na mesma máquina.

## Instalação do Squid

A instalação pode ser feita a partir de várias fontes:

- .rpm
- .deb
- apt
- Código-fonte.

Pode ser instalado a partir de pacotes prontos.

- No entanto, para adicionar alguns recursos, será necessário recompilar o Squid através do arquivo fonte.
- Recomendamos a instalação da última versão estável.

A instalação do Squid pode ser feita a partir de:

- Pacote *.rpm*, utilizado por distribuições RedHat/Fedora;
- Pacote *.deb*, para distribuições Debian;
- Do arquivo fonte, compilando-o localmente;
- Instaladores que se utilizam de repositórios na internet, como é o caso do apt.

Inicialmente pode-se instalá-lo a partir de pacotes prontos; no entanto, deve-se obter também os arquivos fontes, uma vez que para adicionar alguns recursos ao Squid é necessário recompilá-lo. É recomendado ainda instalar a última versão estável.

Considerando que `squid-3.1.20.tar.gz` seja a versão estável, a instalação pode ser efetuada com o comando:

```
# wget http://www.squid-cache.org/Versions/v3/3.1/squid-3.1.20.tar.gz
```

Após essa etapa, será necessário criar um usuário e um grupo para o Squid, procedendo-se em seguida com a compilação por meio dos comandos:

```
# ./configure
# make
# makeinstall
```

Após a instalação, é preciso editar o arquivo de configuração `squid.conf`.

## Configuração do Squid

Arquivo `squid.conf`:

- Possui grande número de parâmetros.
  - Porta, quantidade de memória, localização de arquivos de log.
  - Listas de controle de acesso (ACL).
- É recomendável ir incluindo ACL aos poucos.
- Verificar efetividade de cada parâmetro.



Mais informações podem ser obtidas na página oficial do Squid:  
<http://www.squid-cache.org/>



O arquivo de configuração do squid.conf possui grande número de parâmetros que podem ser utilizados. Ao colocar o Squid em funcionamento pela primeira vez, é recomendável incluir os parâmetros aos poucos, especialmente o ACL, justamente para que se possa perceber a efetividade de cada um. Alguns desses parâmetros:

- **http\_port**: número da porta utilizada pelo servidor, em geral 3128;
- **cache\_mem**: quantidade de memória RAM utilizada pelo proxy web (em MB), default 256 MB;
- **cache\_dir**: define vários parâmetros de cache, como tipo de armazenamento, diretório de cache, quantidade em MB, número de diretórios de primeiro nível, número de diretórios de segundo nível. Exemplo: `cache_dirufs /usr/local/squid/var/cache 500 16 256`;
- **access\_log**: localização do arquivo com logs de acesso ao conteúdo web;
- **cache\_log**: arquivo com informações de log;
- **cache\_store\_log**: arquivo com detalhes sobre objetos armazenados; registra objetos que entraram e saíram, e o tempo de permanência;
- **pid\_filename**: número do processo Squid em execução;
- **visible\_hostname**: nome do host apresentado em mensagens de erro;
- **cache\_effective\_user**: nome dos processos criados pelo Squid;
- **cache\_effective\_group**: grupo do dono dos processos criados pelo Squid;
- **acl REDE 192.168.0.0/255.255.255.0**: exemplo de lista de controle de acesso;
- **http\_accessallow REDE**: informa o que fazer com a lista de controle de acesso.

## Listas de controle de acesso

Access Control List (ACL).

- Regras de acesso utilizadas pelo sistema.
- Controlam quem pode acessar o que e quando.
  - Por meio de um conjunto de regras encadeadas.
- Permitem bloquear ou liberar determinados tipos de acesso.
- Limitam consumo de banda.

As listas de controle de acesso ou Access Control Lists (ACLs) são regras de acesso utilizadas pelo sistema para controlar quem pode acessar o que e quando. Por meio de um conjunto de regras encadeadas, permite bloquear ou liberar determinados tipos de acesso, além de limitar o consumo de banda em determinadas situações.

Entre os parâmetros que podem ser utilizados:

- Origem da requisição;
- Destino da requisição;
- Horário da requisição;
- Endereço MAC;
- Disponibilidade de banda;
- Filtros baseados em strings ou expressões regulares.



Para um bom entendimento do funcionamento das ACLs, é necessário considerar:

- As diretivas do arquivo *squid.conf* são lidas de cima para baixo, com cada solicitação de acesso comparada com cada regra de acesso, até que seja encontrada uma que combine ou até que seja atingido o final do arquivo.
- A última regra deve sempre bloquear todas as solicitações de acesso. Desse modo, caso nenhuma regra prévia corresponda à solicitação efetuada, haverá uma última que bloqueia o acesso.
- Não criar regras redundantes, desnecessárias ou que exijam resolução DNS, para não diminuir o desempenho do proxy.
- Para um grande número de diretivas ou para maior flexibilidade, utilize o Squid associado a outros programas, como o Squirm, SquidGuard ou DansGuardian.

As ACLs possuem uma sintaxe própria, com diretivas para controle e tipos de regras de acesso.

- `acl rede src 192.168.1.0/24`
- `http_accessdeny rede`

Existe um grande número de classes para controle de acesso:

- Hora/data, porta, protocolo etc.
- Número máximo de conexões a partir de um único endereço IP.

ACLs possuem uma sintaxe própria, com diretivas para controle e tipos de regras de acesso. Considere o exemplo:

```
acl rede src 192.168.1.0/24
http_accessdeny rede
```

A primeira linha associa a lista de controle de acesso rede ao intervalo de endereços definido por 192.168.1.0/24.

A segunda linha informa que os elementos da lista de controle de acesso rede devem ter seu acesso bloqueado ao serviço http.

O tipo de uma regra pode ser:

- **scr**: baseada no endereço IP de origem;
- **dst**: baseada no endereço IP de destino;
- **srcdomain**: baseada no domínio do cliente que solicita o acesso;
- **dstdomain**: baseada no domínio do servidor de destino da solicitação;
- **srcdomain e dstdomain**: tipos que solicitam resolução DNS e devem ser evitados.

Existe um grande número de classes para controle de acesso por:

- Hora e data;
- Porta;
- Protocolo;
- Métodos get ou post;
- Tipo de navegador;
- Usuário logado;



- Tipo de conteúdo (MIME type) solicitado através de cabeçalho content-type;
- Número máximo de conexões a partir de um único endereço IP.

## Exemplos de ACLs

Lista de acesso users permite o acesso durante o horário do almoço de segunda a sexta, no horário das 12:00 às 13:55, negando acesso em outras ocasiões.

```
acluserssrc 192.168.1.0/24
aclalmoco time MTWHF 12:00-13:55
http_accessallowusersalmoco
http_accessdenyusers
```

### Exemplo 1

ACL com controle de data e hora. A lista de acesso users permite o acesso durante o almoço de segunda a sexta, no horário de 12h até 13h55, negando acesso nos demais horários.

```
acl users src 192.168.1.0/24
aclalmoco time MTWHF 12:00-13:55
http_access allow users almoco
http_access deny users
```

### Exemplo 2

ACL com controle de tipos de arquivos, negando acesso a arquivos *.mp3*, *.avi*, *.zip* e *.exe*. Pode ser útil para impedir o download de músicas, vídeos e executáveis.

```
acl mp3 url_regex -i.*\.mp3$
aclaviurl_regex -i.*\.avi$
acl zip url_regex -i.*\.zip$
acl exeurl_regex -i.*\.exe$
http_access deny all mp3
http_access deny all avi
http_access deny all zip
http_access deny all exe
http_access allow all
```

## Configuração dos navegadores

Navegadores suportam configuração de proxy.

No navegador Mozilla Firefox:

- Menu ferramentas.
- Opções.
- Menu Avançado.
- Rede.





- Determine como o Firefox conecta-se à internet.
- Configuração manual do proxy.
- Informar o endereço IP e porta utilizados.



Qualquer um dos navegadores disponíveis pode ser configurado para acesso via proxy. De modo geral, por meio de uma interface de configuração, é possível informar os dados do servidor, como endereço IP e porta.

- **Mozilla Firefox:** deve-se acessar o menu “Ferramentas”, “Opções”, “Menu Avançado”, “Rede”, determinar como o Firefox conecta-se à internet, fazer a configuração manual do proxy e, finalmente, informar o endereço IP e porta utilizados.
- **Internet Explorer:** entre no menu “Ferramentas”, “Opções da internet”. Selecione a aba “Conexões” e clique no botão “Configurações” da LAN. Na janela “Configurações” da rede local (lan), na seção “Servidor proxy”, marque a opção para usar um servidor proxy e informe o endereço IP e a porta utilizada pelo servidor proxy.

## Proxy transparente

Resolve dois problemas:

- Necessidade de configuração de cada navegador instalado na rede.
- Controle de usuários mais experientes que removem as configurações do proxy, evitando assim o controle de tráfego.



A utilização de proxy transparente resolve dois problemas. O primeiro é a necessidade de configuração de cada navegador instalado na rede. O segundo ocorre quando um usuário com um pouco mais de experiência remove as configurações do proxy, evitando o controle de tráfego.

Desse modo, a utilização de proxy fica oculta, e os usuários usam-no mesmo que não queiram. Facilita ainda a implementação da política de segurança, pois tira do navegador a configuração para acesso ao proxy. Todas as solicitações feitas na porta 80 são redirecionadas para o Squid, que pode então controlar o tráfego por meio das ACLs.

Para configuração do proxy transparente são necessárias três etapas:

- Recompilar o kernel para adicionar suporte ao proxy transparente;
- Recompilar o Squid para adicionar suporte ao proxy transparente;
- Incluir uma regra no firewall iptables.

Para recompilar o kernel é necessário marcar em “Opções gerais”:

- Em “Networking support”: marcar “Sysctl support”.
- Em “Networking options”: marcar “Network Packet Filtering” e “TCP/IP Networking”.
- Em “Networking options”, opção “IP: Netfilter Configuration”: marcar “Connection tracking”, “IP tables support”, “Full NAT” e “REDIRECT target support”.
- Em “File Systems”: marcar “/proc filesystem support”.



Antes de recompilar o kernel, carregue as configurações em vigor e depois efetue as modificações mencionadas.

Na configuração são necessárias três etapas para adicionar suporte ao proxy transparente:



- Recompilação do kernel.
- Recompilação do Squid.
- Inclusão de uma regra no firewall iptables.

Em seguida, recompile o kernel com os comandos:

- `make clean;`
- `makedep;`
- `makebzImage;`
- `make modules;`
- `makemodules_install.`

Após a recompilação do kernel, é necessário recompilar o Squid para adicionar suporte ao proxy transparente, o que pode ser feito a partir do diretório com os arquivos fontes do Squid, através dos seguintes comandos:

```
# ./configure --enable-linux-netfilter
# make
# make install
```

Por último, deve-se incluir uma regra no firewall iptables para redirecionar o tráfego da porta 80 para o Squid. A regra é:

```
# iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j
REDIRECT --to-port 3128
```

Onde 3128 é a porta do Squid.

## Redirecionadores

Ferramentas que permitem ao administrador da rede redirecionar determinadas páginas acessadas pelos usuários.



Normalmente em dois casos:

- Desvio de downloads.
- Advertência de usuários.

Os redirecionadores de URL são ferramentas que permitem ao administrador da rede redirecionar determinadas páginas acessadas pelos usuários, o que normalmente é feito em dois casos:

- Desvio de downloads;
- Advertência de usuários.

Esses redirecionadores adicionam recursos ao Squid, que já tem entre suas funcionalidades o bloqueio a palavras e páginas proibidas.

No primeiro caso o administrador, após observar grande volume de downloads de um ou mais arquivos a partir da internet, pode optar por disponibilizar esse arquivo localmente e redirecionar todos os pedidos, economizando volume de tráfego no link de internet.

Um segundo exemplo ocorre quando o usuário acessa páginas com conteúdo proibido. Nesse caso, o acesso é redirecionado para uma página com advertências. Entre os redirecionadores, podemos citar:

- Jesred;
- Squirm;
- SquidGuard;
- DansGuardian.

A seguir, abordaremos a instalação e a utilização do Squirm.

Entre os redirecionadores, podemos citar:

- Jesred.
- Squirm.
- SquidGuard.
- DansGuardian.

O Squirm é um redirecionador desenvolvido para trabalhar em conjunto com o Squid, de fácil instalação e recomendado na documentação do Squid. Para instalá-lo, é recomendado efetuar o download na página do desenvolvedor. Em seguida deve-se desempacotar o arquivo:

```
# tar -xvfsquirm-versao.tar
```

Após desempacotar o Squirm e antes de compilá-lo, deve-se compilar a biblioteca regex.o:

```
# cd regex
# ./configure
# make clean
# make
# cp -p regex.oregex.h
```

No diretório do Squirm:

```
# cd ..
# make
# makeinstall
```

É necessário ainda modificar o *owner* do executável do Squirm, configurando o mesmo do Squid:

```
# chownsquid.squid /usr/local/squirm/bin/squirm
```

No Squirm, devem ser editados os dois arquivos de configuração:

- squirm.local
  - Endereços das redes locais.
- squirm.pattern
  - Regras a serem utilizadas:

```
regexi^http://www\.porno\.com\.br/. *http://www/advertencia.htm
```

Para a configuração do Squirm, devem ser editados os dois arquivos de configuração:

- *squirm.local*
- *squirm.pattern*

No arquivo *squirm.local* devem ser incluídos os endereços das redes locais. Por exemplo:

- 127.0.0
- 192.168.0

No arquivo *squirm.pattern* devem ser incluídas as regras utilizadas, no seguinte formato: `regex|regexi<padrão><redirecionamento>`.

Por exemplo:

```
regexi ^http://www\.porno\.com\.br/.* http://www/advertencia.htm
```

Onde `advertencia.htm` é a página para a qual os usuários são redirecionados.

Podemos adicionar várias regras, onde cada uma pode redirecionar para uma página diferente. O Squirm é desnecessário quando se utiliza iptables com NAT.

Caso a necessidade seja ainda de utilização para redirecionamento associado a bloqueio de conteúdo, pode-se utilizar as ferramentas SquidGuard ou DansGuardian, pois além do redirecionamento, elas podem ainda efetuar bloqueio de acesso baseado em listas negras. A mesma solução pode ser obtida com Squid e Squirm.





## Roteiro de Atividades 9

Esta atividade pode ser feita tanto pelo cliente Windows quanto pelo cliente Linux.  
É importante que tanto a máquina cliente quanto o servidor estejam na mesma rede.  
Teste o funcionamento da rede através do comando *ping*.



Para realizar este Roteiro de Atividades, é necessário que o servidor esteja configurado com duas placas de rede: uma delas deve possuir rota e acesso à internet.

Vamos supor que a nossa interface *eth0* será a interface com acesso à internet, configurada pelo DHCP do laboratório.

1. No servidor, ative o *ip\_forward*, editando o arquivo */etc/sysctl.conf* e inserindo a linha a seguir:

```
net.ipv4.ip_forward=1
```

### Atividade 9.1 – Instalação e configuração do servidor proxy Squid

Nesta atividade, será feita a instalação a partir do repositório Debian. Observe que serão instalados os pacotes *iptables* (*firewall*), *SARG* (gerador de relatório de acesso) e *apache2* (servidor web para publicação dos relatórios do SARG).

Caso o SARG não esteja disponível na lista padrão de repositórios, devemos incluir um novo repositório conforme orientação do instrutor. Em seguida, efetue a instalação dos pacotes *squid*, *iptables*, *sarg* e *apache2*.

1. As configurações a seguir deverão estar descritas no arquivo */etc/squid/squid.conf* com as modificações necessárias de acordo com a característica de cada rede local.

Cada comando recebeu um comentário para facilitar o entendimento. Ainda assim, não deixe de ler o manual do Squid para obter mais informações.

```
#Porta utilizada pelo Cliente
http_port 3128

#Porta utilizada para troca de informações entre Proxies
icp_port 3130

# nome do proxy
visible_hostname cache.empresa.com.br

#Manter configuração Default
acl QUERY urlpath_regex cgi-bin \?

no_cache deny QUERY

#Configurações do Cache

cache_mem 96 MB

cache_swap_low 75
```



```

cache_swap_high 95
maximum_object_size 900 MB
minimum_object_size 0 KB
maximum_object_size_in_memory 32 KB
#Configurações do diskd
cache_dir ufs /var/spool/squid 20000 64 256
#path dos Logs
cache_access_log /var/log/squid/access.log
cache_log /var/log/squid/cache.log
cache_store_log none
# definir o rotate log
logfile_rotate 10
#não emula o log do Apache
#emulate_httpd_log on
#opções de FTP
ftp_user admin@empresa.com.br
ftp_passive on
#Manter as configurações Default
refresh_pattern ^ftp:          1440  20%  10080
refresh_pattern ^gopher:      1440  0%   1440
refresh_pattern .              0     20%  4320
#acl defaults
acl all src all
acl manager proto cache_object
acl localhost src 127.0.0.1/32
acl to_localhost dst 127.0.0.0/8 0.0.0.0/32
acl SSL_ports port 443 563
acl Safe_ports port 80
acl Safe_ports port 21
acl Safe_ports port 443 563
acl Safe_ports port 70
acl Safe_ports port 210

```

```

acl Safe_ports port 1025-65535
acl Safe_ports port 280
acl Safe_ports port 488
acl Safe_ports port 591
acl Safe_ports port 777
acl CONNECT method CONNECT

#Acesso à página do cache manager
http_access allow manager localhost
http_access deny manager

#Acesso Defaults
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access allow SSL_ports

#ACLS e regras de acesso

# Permitir o acesso ao servidor proxy a partir da rede interna
acl rede src 192.168.1.0/255.255.255.0
http_access allow rede

#regra default
http_access deny all

#Configuracoes defaults
http_reply_access allow all
icp_access allow all
miss_access allow all
never_direct allow all

```

2. Parar o serviço do squid:

```
# /etc/init.d/squid stop
```

3. Criar a árvore de diretórios de cache:

```
# squid -z
```

4. Reiniciar o serviço:

```
# /etc/init.d/squid start
```



## Atividade 9.2 – Configuração dos navegadores

---

Nesta atividade, será feita a configuração do navegador Internet Explorer. Essa configuração deverá ser feita em todas as máquinas de usuário que acessarão a internet por meio do servidor proxy squid.

Configure o navegador para acessar a internet passando pelo servidor Proxy.

## Atividade 9.3 – Configuração de listas de controle de acesso

---

Nesta atividade, será configurado o proxy squid com a utilização de listas de controle de acesso por endereço por MAC, IP, hora, tipo de arquivo e sites restritos.

O Squid avalia as regras de acesso por procedência, ou seja, a primeira regra com a qual a solicitação se adequar será a regra aplicada pelo servidor. Esse comportamento faz com que todas as regras listadas a seguir devam anteceder a linha *http\_access deny all* do arquivo de configuração.

- Limitar o acesso à internet para o endereço MAC da estação Windows: procure pela diretiva *arp*;
- Limitar o acesso à internet para o endereço IP do cliente Windows: procure pela diretiva *src*;
- Não permitir o acesso à internet nos horários de 0h as 6h e de 19h as 23:59h, durante os dias da semana;
- Proibir o download de arquivos com extensão *.mp3* e *.avi*;
- Restringir o acesso ou pesquisa de algumas palavras, como por exemplo: *sexo*, *playboy*, *sexy* etc.;
- Proibir o acesso a uma lista de sites.

## Atividade 9.4 – Configuração do SARG

---

SARG é um gerador de relatório de acesso e uso de internet. Com essa ferramenta é possível identificar e controlar os sites que os usuários estão acessando durante o dia de trabalho.

Configure o SARG para que seja possível visualizar o relatório. Fique atento aos parâmetros *output\_dir* e *access\_log*.

## Atividade 9.5 – Proxy transparente

---

Altere a configuração do servidor Proxy para que os clientes possam realizar o acesso à internet de forma transparente. Lembre-se de remover a configuração de proxy dos navegadores dos clientes.

Precisaremos criar uma regra no iptables para realizar o redirecionamento dos pacotes.

Exemplo:

```
# Redirecionar os pacotes da porta 80 para porta 3128
iptables -I PREROUTING -t nat -p tcp -s 192.168.1.0/24 --dport 80 -j
REDIRECT --to-port 3128

#Ativar o serviço de NAT
iptables -t nat -I POSTROUTING -s 192.168.1.0/24 -j MASQUERADE
```

# 10

## Servidor Samba

### objetivos

Instalar e configurar o Samba, instalar o SWAT e o Servidor Primário de Domínio ou Primary Domain Controller (PDC), controlar informações de usuários e autenticação para clientes Windows.

O servidor Samba, compartilhamento de um disco Linux, acesso ao Windows do Linux, montagem de diretórios do Windows e configuração do Samba com SWAT.

### conceitos

### Introdução ao Samba

- Samba e projetos relacionados.
- Instalação e configuração do servidor Samba.
  - ▣ Seção (homes).
  - ▣ Compartilhamento de um disco Linux.
  - ▣ Acesso ao Windows do Linux.
  - ▣ Montagem de diretórios do Windows.
  - ▣ Configuração do Samba com SWAT.
  - ▣ Servidor primário de domínio.



O estudo deste capítulo se inicia pelos conceitos do serviço Samba, abordado desde suas origens, passando por aspectos técnicos importantes, e concluindo com sua instalação, configuração e testes relevantes nas situações reais em que está inserido.

Na primeira parte será apresentada a teoria e projetos relacionados ao Samba.

Em seguida, estudaremos a instalação e a configuração do Samba, com especial atenção aos parâmetros mais importantes e à seção (homes) de seu arquivo de configuração. Apresentaremos ainda o compartilhamento de um disco Linux, o acesso ao Windows a partir do Linux e a montagem de diretórios do Windows.

Na sequência apresentaremos o SWAT, uma interface web para administração do Samba. Na etapa seguinte veremos o Servidor Primário de Domínio ou Primary Domain Controller (PDC), e como colocar o Samba para atuar como PDC, controlando as informações sobre usuários e autenticação para clientes Windows.



Atividades práticas:

- Configuração do servidor Samba;
- Verificação do funcionamento do servidor Samba;
- Configuração de uma máquina Windows para autenticar no Samba;
- Adicionar e remover usuários;
- Compartilhamento de diretório;
- Utilização do SWAT.

## Samba

- Conjunto de aplicações baseadas no protocolo Server Message Block (SMB).
- Possibilita a comunicação com todas as estações da rede que utilizam SMB:
  - Família Windows.
  - XENIX-Net.
  - IBM Lan Server.
  - 3COM's3 + OPEN.



O Samba é um conjunto de aplicações baseadas no protocolo Server Message Block (SMB), possibilitando a comunicação com todas as estações da rede que utilizam SMB, como estações WinXP, WinNT, Vista, Win7 e Win8. Outros Sistemas Operacionais que utilizam o protocolo SMB: XENIX-Net (SCO UnixWare), IBM Lan Server e 3COM's3 + OPEN.

As ferramentas do Samba permitem que máquinas Linux compartilhem unidades de disco e impressoras com servidores ou estações Windows. Com o Samba, todas as máquinas Linux de uma rede podem ser vistas como uma ou mais pastas lógicas para os usuários do Windows. Uma máquina Windows utilizando TCP/IP e NetBEUI visualiza os recursos do Samba em máquinas Linux da mesma maneira que em uma rede Windows. Com a ferramenta smbfs, as máquinas Linux podem se conectar aos recursos compartilhados em máquinas Windows (diretórios e impressoras). Um cliente Linux visualiza um servidor de arquivos SMB como um sistema de arquivos Linux.

- Ferramentas Samba permitem que máquinas compartilhem unidades de disco e impressoras, independente do Sistema Operacional.
- Ferramenta smbfs.
  - Máquinas Linux podem usar os recursos compartilhados em máquinas Windows.



## Terminologia

- **Server Message Block (SMB):** protocolo de compartilhamento de arquivos e impressoras da Microsoft;
- **Common Internet File System (CIFS):** por volta de 1996, a Microsoft decidiu que o SMB necessitava da palavra "internet"; então, passou a ser chamado de CIFS;
- **Direct-Hosted:** método de prover compartilhamento de serviços sob a porta 445/TCP, usando apenas DNS para resolução de nomes em vez de WINS;
- **Inter-Process Communication (IPC):** método para comunicar informações específicas entre programas;

- **Network Basic Input/Output System (NetBIOS):** não é um protocolo; é um método de comunicação através de um protocolo existente, que controla as funções essenciais do tráfego de entrada/saída via rede. Deve ficar claro que NetBIOS é um padrão de transporte, não um protocolo;
- **NetBIOS Extended User Interface (NetBEUI):** é um protocolo, não um padrão. Não é roteável, ou seja, o tráfego no lado de um roteador será incapaz de se comunicar com o outro lado;
- **NetBIOS over TCP (NBT):** permite o uso continuado do tráfego NetBIOS sobre TCP/IP. Como resultado, nomes NetBIOS são semelhantes a endereços IP, e tipos de nomes NetBIOS são conceitualmente equivalentes a portas TCP/IP. É assim que acontece o compartilhamento de arquivos e diretórios em Windows, que se baseiam em três portas: NetBIOS Name Service (nbname) via UDP porta 137, NetBIOS Datagram Service via UDP porta 138 e NetBIOS Session Service (nbssession) via TCP porta 139. A resolução de nomes é feita via WINS, broadcasts NetBIOS e DNS.

## Samba e projetos relacionados

Projetos diretamente relacionados ao Samba:

- **Server Message Block File System (SMBFS).**
  - ▣ Permite a montagem de compartilhamentos SMB.

Projetos diretamente relacionados ao Samba:

- **Common Internet File System Virtual File System (CIFS VFS).**
  - ▣ Sucessor do SMBFS, provendo funcionalidades avançadas de sistemas de arquivos de rede.
  - ▣ Inclui suporte a DFS.
  - ▣ Segurança por sessão de usuário.
  - ▣ Caching distribuído.

Há dois projetos de sistemas de arquivos para Linux diretamente relacionados ao Samba: SMBFS e CIFS VFS. Ambos estão disponíveis no próprio kernel do Linux.

Server Message Block File System (SMBFS) permite que façamos a montagem de compartilhamentos SMB e tenhamos acesso como em qualquer outro diretório, o que é útil se desejamos montar sistemas de arquivos sem um servidor SMBFS.

Tradicionalmente, SMB usa porta 137 UDP (NetBIOS name service ou netbios-ns), porta 128 UDP (NetBIOS datagram service) e porta 139 TCP (NetBIOS session service).

Em geral, sessões SMB são estabelecidas na seguinte ordem:

- **TCP Connection:** estabelecimento de conexão “TCP 3-way handshake” para a porta 139/TCP ou 445/UDP;
- **NetBIOS Session Request:** define o nome NetBIOS da máquina local e o nome do servidor a ser acessado;
- **SMB Negotiate Protocol:** determina o protocolo a usar, que será um dos seguintes: PC Network Program 1.0 (Core) e Microsoft Networks 1.03 (Core Plus), apenas para o modo de nível de segurança de compartilhamento, entre outros;
- **SMB Session Startup:** senhas são encriptadas (ou não) de acordo com um dos seguintes métodos: Null (nenhuma encriptação); Cleartext (nenhuma encriptação); LM e NTLM; NTLM; NTLMv2;



- **SMB Tree Connect:** conexão a um nome compartilhado (por exemplo, `\\servername\share`), conexão a um tipo de serviço (por exemplo, `IPC$` named pipe).

Common Internet File System Virtual File System (CIFS VFS) é o sucessor para o SMBFS. A intenção desse módulo é prover funcionalidades avançadas de sistemas de arquivos de rede, incluindo suporte a Distributed File System (DFS) segurança por sessão de usuário, caching distribuído seguro (oplock), entre outros.

Há outras implementações open source de clientes CIFS, como jCIFS Project, que provê um toolkit SMB cliente escrito em Java.

## Instalando o Samba

Pacotes:

- **samba-common:** componentes comuns ao servidor e clientes.
- **Samba:** componentes do servidor.
- **samba-client:** para que um cliente Linux acesse uma máquina Windows.

Compilação:

- É necessário especificar a localização de `sbm.conf`.
- Compile com `make`.
- Verifique se os daemons `smbd` e `nmbd` estão rodando.

Muitas distribuições já incluem todos os pacotes Samba. Caso uma determinada distribuição não possua tais pacotes, é necessário instalá-los manualmente através de `rpm` ou `apt-get`. Geralmente, são instalados três pacotes:

- **samba-common:** instala os componentes compartilhados tanto pelo pacote `samba-client` quanto pelo pacote `samba`;
- **samba:** instala os componentes do servidor;
- **samba-client:** possibilita que um cliente Linux acesse uma máquina Windows.

A versão mais recente de Samba está disponível no formato de código-fonte. Após executar o download do código-fonte, é necessário ler o arquivo *Readme*. O arquivo *docs/Samba3-HOWTO.pdf* descreve como executar a instalação a partir do código-fonte.

O próximo passo envolve a configuração do Samba de acordo com o seu Sistema Operacional. Para configurar programas fonte de acordo com seu Sistema Operacional, será necessário criar o script `configure`. Para criá-lo execute os seguintes comandos:

```
# cd samba-X.X.X/source3
# ./autogen.sh
```

Para montar os binários, execute o programa `./configure`. Isso vai configurar o Samba automaticamente para o seu sistema.

```
# ./configure
```

Para compilar, execute o comando `make`.

```
# make
```

Uma vez que a compilação ocorreu sem erro, execute o comando `make install` para instalar os binários e os manual do Samba.



```
# make install
```

Após a instalação, é necessário verificar se os daemons *smbd* e *nmbd* estão rodando.

Vale ressaltar que o servidor Samba é composto por dois ou três daemons:

- **nmbd**: manipula todos os pedidos de registro e resolução de nomes e deve ser o primeiro comando inicializado como parte do processo de inicialização do Samba.
- **smbd**: manipula todos os serviços baseados em TCP/IP para operações em arquivos e impressão, além de gerenciar autenticações locais; deve ser inicializado após *nmbd*.
- **winbindd**: deve ser inicializado quando o Samba é membro de um domínio Windows NT4 ou ADS, e também quando o Samba tem relações com outros domínios, com a função de verificar o arquivo *smb.conf*, procurando a presença dos parâmetros *idmap uid* e *idmap gid*.

Principais arquivos:

- **smbclient**: cliente SMB para máquinas Linux.
- **smbstatus**: mostra as conexões ativas na máquina local.
- **/etc/samba/smb.conf**: arquivo de configuração do Samba.
- **smbcontrol**: envia mensagens para os daemons *nmbd*, *smbd* e *winbindd*.
- **smbutil**: interface de controle para requisições.
- **smbspool**: envia um arquivo para uma impressora associada ao SMB.
- **smbpasswd**: altera a senha dos usuários do Samba.

Os daemons estão localizados no diretório */usr/sbin*. Os daemons SMB */usr/sbin/smbd* e */usr/sbin/nmbd* iniciam a partir do diretório */etc/rc.d/init.d* no Red Hat Linux.

A seguir, apresenta-se uma breve descrição dos principais arquivos em uma instalação Samba. Os arquivos binários são armazenados em */usr/bin*.

- **smbclient**: cliente SMB para máquinas Linux. Permite que uma máquina Linux acesse um servidor SMB, através de uma interface do tipo FTP;
- **smbstatus**: mostra as conexões ativas na máquina local;
- **/etc/samba/smb.conf**: arquivo de configuração do Samba;
- **smbcontrol**: envia mensagens para os daemons *nmbd*, *smbd* e *winbindd*;
- **smbutil**: interface de controle para requisições;
- **smbspool**: envia um arquivo para uma impressora associada ao SMB;
- **smbpasswd**: altera a senha dos usuários do Samba.

Como testar o arquivo de configuração? Para validar o conteúdo de *smb.conf*, use o programa *testparm*. Se *testparm* executar corretamente, listará todos os serviços carregados. Se não, dará uma mensagem de erro. Use o comando:

```
# testparm /etc/samba/smb.conf
```



Execute sempre esse teste, em situações de mudanças de configuração do *smb.conf*.



# Configurando o Samba

Parâmetros globais.

- Configurar variáveis que o Samba utiliza.
- Cada seção representa um compartilhamento no servidor.

Exemplo de smb.conf

```
[global]
workgroup = WKG
netbios name = MYNAME
[share1]
path = /tmp
[share2]
path = /my_shared_folder
comment = Some random files
```



**Figura 10.1**  
Exemplo 'smb.conf'.

Todas as seções do arquivo *smb.conf* iniciam com um cabeçalho de seção delimitado por [] — [global], [homes], [printers] etc. Na seção [global], são configuradas as variáveis que o Samba utiliza para definir a forma como devem ser compartilhados todos os recursos.

Cada seção em *smb.conf* representa um compartilhamento no servidor Samba. A seção “global” é especial, já que contém configurações que se aplicam ao servidor Samba completo e não a um compartilhamento particular.

## Descrição dos principais parâmetros

- **workgroup**: o nome do grupo de trabalho Windows ou do nome do domínio WinNT. Se o nome é “Grupo”, um sistema Windows verá esse nome na rede;
- **server string**: fornece uma descrição do servidor Samba que será visível a todas as máquinas Windows. Escreva um texto que descreva o tipo de servidor que possuímos;
- **printcap name**: especifica onde está localizado o arquivo de definição de impressoras do Linux. Todas as definições de impressoras no Linux podem ser encontradas nesse arquivo;
- **load printers**: indica para o Samba que ele deve tornar as impressoras locais acessíveis para qualquer cliente SMB. Se esse parâmetro está setado, não é necessário configurar as impressoras individualmente.

## Parâmetros globais

- workgroup.
- server string.
- printcap name.
- load printers .
- printing.
- log file.
- security.
- dns proxy.



Descrição dos principais parâmetros:

- **printing**: seleciona o sistema de impressão utilizado; normalmente não é necessário alterar o valor padrão;
- **log file**: indica o nome do arquivo que deve ser utilizado para armazenar as informações de log. Por padrão, é criado um arquivo de log para cada cliente que acessa o servidor Samba;
- **security**: é um dos parâmetros mais importantes da seção Global. Ele indica como o Samba deve administrar a autenticação dos clientes. Existem quatro variantes: *share*, *user*, *server* e *domain*. Se for selecionado *user*, o servidor Samba faz uma requisição ao cliente para fornecer um usuário e uma senha para autenticação. Essa solução é mais adequada quando os usuários possuem o mesmo nome e senha, tanto no Windows quanto no Linux. Se for selecionado *share*, o servidor Samba requisita uma senha para cada recurso compartilhado. Se for selecionado *server* ou *domain*, o servidor Samba requisita um usuário e uma senha do cliente, e faz a autenticação utilizando outro servidor SMB. Para isso, é necessário setar o parâmetro *password server*, que indica o nome do servidor SMB que deve ser utilizado para autenticação;
- **dns proxy**: se estiver setado como "yes", o daemon *nmbd* tentará tratar os nomes de NetBIOS como nomes de domínios de internet, através do protocolo DNS. Normalmente, essa opção fica desabilitada.

## Seção [homes]

- Permite compartilhar os diretórios home dos usuários automaticamente.
- Ao se logar, um usuário Windows receberá como seu diretório home o diretório compartilhado pelo Samba.
- Permite aos usuários de ambas as plataformas, Linux e Windows, possuírem apenas um diretório home, acessível a ambos os sistemas.



**Figura 10.2**  
Exemplo da  
seção *homes*.

```
Exemplo
[homes]
comment = Diretórios pessoais dos usuários
netbios name = MYNAME
browseable = no
writable = yes
```

Se um cliente tenta se conectar a um recurso compartilhado que não existe no arquivo *smb.conf*, o Samba procurará pela seção [homes] de compartilhamento no seu arquivo de configuração. Se a seção existir, o nome do recurso compartilhado será entendido como um nome de usuário Linux, o qual será usado para testar a sua existência no banco de dados do servidor Samba. Se existir, o Samba assume que o cliente é um usuário Linux tentando se conectar ao seu diretório home no servidor.

Por exemplo, imagine que um computador cliente esteja se conectando a um servidor Samba pela primeira vez, e tenta se conectar a um compartilhamento de nome [maria]. No arquivo *smb.conf* não existe compartilhamento com esse nome, mas existe a seção [homes], então o Samba procura por "maria" no arquivo *passwd* do Linux e encontra o registro "maria". O Samba então valida a senha informada pelo cliente, no arquivo *passwd* se ela não estiver criptografada, ou no arquivo *smbpasswd* se senha criptografada estiver em uso. Se a senha informada estiver correta, então o Samba sabe que ela tem direito de acesso e criará um compartilhamento de nome [maria].





A seguir a configuração da seção [homes], que permite que os usuários Windows acessem seus diretórios do Linux:

```
[homes]

comment = Diretórios pessoais dos usuários

browseable = no

writable = yes
```

A máquina Linux precisa fazer parte da LAN para que os usuários Windows possam acessar seus diretórios compartilhados. Desta forma, um usuário local pode se conectar a um disco rígido do servidor Samba pelo Windows Explorer.

## Compartilhamento de um disco Linux

Alguns parâmetros:

- adminusers.
- comment.
- copy.
- create mode.
- browseable.
- Entre muitos outros.



|                                                                                                                                |                                                                                                                                                   |
|--------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Exemplo 1</p> <pre>[public] comment = Diretório Público path = /home/data public = yes writable = yes printable = yes</pre> | <p>Exemplo 2</p> <pre>[public] comment = Diretório Público path = /home/data public = yes writable = yes printable = no write list = @grupo</pre> |
|--------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|

**Figura 10.3**  
Exemplos de seção public.

A configuração do arquivo *smb.conf* para compartilhar um disco Linux é relativamente simples.

- O Exemplo 1 apresenta alguns parâmetros de configuração de seção [public].
- O Exemplo 2 mostra o que deve ser feito para tornar o diretório acessível a todos, mas com permissão de escrita somente para um determinado grupo.

Alguns parâmetros específicos dos compartilhamentos:

- **adminusers:** lista de usuários que possuirão privilégios administrativos do compartilhamento;
- **browseable:** parâmetro booleano ("yes", "no") que indica se o compartilhamento será visível na lista de compartilhamentos disponíveis;
- **comment:** informa uma descrição do compartilhamento;
- **copy v:** permite clonar compartilhamentos, isto é, o compartilhamento foi duplicado, e os novos parâmetros adicionados revogam os parâmetros do compartilhamento que foi copiado;
- **create mode:** define com quais permissões os arquivos serão criados no compartilhamento.



Para parâmetros adicionais, consulte o site <http://www.samba.org>

## Como acessar o Windows do Linux



- Duas maneiras.
- Executando o programa smbclient.
- Montando um diretório lógico do Windows com um sistema de arquivos local.
- Usando smbclient.
- Para visualizar diretórios, serviços e servidores disponíveis no servidor.
- Para acessar um recurso disponível no servidor.
- Para enviar mensagens para o servidor.

Há duas maneiras de acessar dados de um servidor Windows a partir do Linux. Na primeira, deve-se executar o programa smbclient, que possui uma interface FTP na linha de comando. Na segunda pode-se montar um diretório lógico do Windows com um sistema de arquivos local.

Antes de utilizar o smbclient, é necessário que a máquina Windows possua algum compartilhamento. Lembre-se de atualizar o arquivo `/etc/hosts` com os endereços IP e os nomes de todas as máquinas da rede, caso a máquina não acesse um servidor DNS.

Com o smbclient, é possível visualizar os diretórios que são acessíveis em um certo servidor, além dos serviços disponíveis, dos servidores acessíveis e dos grupos de trabalho ou domínios visíveis, da seguinte forma:

```
# smbclient -L <servidor>
```

Para acessar um recurso disponível em um determinado servidor, utilize o seguinte comando:

```
# smbclient //<servidor>/<recursos> [<senha>]
```

Por exemplo, se desejar acessar uma pasta compartilhada em uma máquina Windows chamada "tunisia", a sintaxe é:

```
# smbclient //tunisia/Documentos
```

O parâmetro -M permite o envio de mensagens para o servidor. Sua sintaxe é:

```
# smbclient -M <servidor>
```

## Montagem de diretórios do Windows



É necessário que todas as ferramentas do Samba estejam instaladas, inclusive o suporte a *smbfs*.

Comandos:

- Montagem:

```
mount -t smbfs //<servidor>/<recurso> <ponto de montagem> <opções>
```

- Desmontagem:

```
umount <ponto de montagem>
```

Para acessar um servidor Windows de um cliente Linux, por montagem, é necessário que todas as ferramentas do Samba estejam instaladas, e o kernel do Linux deve estar com o suporte a *smbfs* compilado.



A sintaxe padrão do comando *mount* é:

```
# mount -t smbfs //tunisia/Documents /mnt/windocs
```

Tunisia é o nome do servidor Windows e */mnt/windocs* deve existir antes da execução de *smbmount*. O comando *smbmount* utiliza o comando *mount* com a opção *-t smbfs* para executar suas operações.

É também possível especificar um usuário e uma senha:

```
# mount -t smbfs -o username=usuario,password=123456 //tunisia/
Documents /mnt/windocs
```

Para desmontar um diretório, pode-se utilizar o seguinte comando:

```
# umount <ponto de montagem>
```

## Configuração do Samba com SWAT

Samba Web Administration Tool (SWAT).

- Utilitário de configuração que pode ser acessado, via web, no endereço <http://localhost:901>
  - ▣ Basta fornecer a senha de root para acessar.
- Logins e senhas devem ser os mesmos que os usuários utilizarão para se logar no Windows.

O Samba pode ser configurado com o SWAT, um utilitário de configuração acessível via web, no endereço <http://127.0.0.1:901>. Basta fornecer a senha de root para acessá-lo.

Antes devem ser criados logins para todos os usuários que forem acessar o servidor; por sinal, os logins e senhas devem ser os mesmos que os usuários utilizarão para se logar no Windows.

Ao abrir o SWAT, pode-se observar vários links para a documentação disponível sobre o Samba, que pode ser consultada para obtenção de detalhes sobre o sistema.

Samba Web Administration Tool (SWAT).

- Terminadas as configurações:
  - ▣ O servidor aparecerá no ambiente de redes.
  - ▣ Os compartilhamentos podem ser acessados de acordo com as permissões que tiverem sido configuradas.
  - ▣ Podem ser mapeados como unidades de rede.

Acesse primeiro a seção “Password”, onde deverão ser cadastrados todos os usuários que terão acesso às pastas compartilhadas através do Samba. Em seguida, acesse a seção “Globals”, que engloba todas as configurações de rede e de acesso.

Nas opções Workgroup e NetBios name, identifique o nome do computador e de seu respectivo grupo de trabalho.

Na seção “Security”, indique “User”, o que permitirá definir os usuários que terão acesso ao sistema.

A opção “Hosts Allow” deve incluir os endereços IP de todos os computadores que terão permissão para acessar o servidor. Para liberar acesso a todos os IPs da rede, basta indicar a primeira parte do IP (netid). A opção “Hosts Deny”, ao contrário, especifica máquinas dentro do escopo configuradas na opção “Hosts Allow”, que não terão permissão para acessar o servidor.

As pastas a serem compartilhadas com as estações podem ser configuradas através da seção “Shares”. Para criar um compartilhamento, basta escrever seu nome no campo no topo da tela e clicar no botão “Create Share”. Depois de criado um compartilhamento, escolha-o na lista e clique no botão “Choose Share” para configurá-lo. O campo “Path” indica qual pasta será compartilhada.

! Os usuários só terão permissão para acessar pastas que o login permite acessar.

Terminadas as configurações, o servidor aparecerá no ambiente de redes, como se fosse um servidor Windows. Os compartilhamentos podem ser acessados de acordo com as permissões que tiverem sido configuradas e podem ser mapeados como unidades de rede, entre outros recursos.

## Servidor primário de domínio

- Ideia que simplifica tarefas administrativas da rede.
- Único servidor armazena informações sobre usuários e autenticação.
- Perfis armazenados no controlador de domínio.
- Controlador de domínio gerencia todas as requisições de autenticação.



A ideia de controlador de domínio primário simplifica várias tarefas administrativas da rede, pois permite que um único servidor armazene as informações sobre os usuários de um domínio e forneça autenticação para esses usuários. Os perfis dos usuários são armazenados no controlador de domínio, que gerencia todas as requisições de autenticação, permitindo que os usuários acessem diferentes recursos da rede, sem necessitar de múltiplas autenticações.

As seguintes funcionalidades são implementadas quando se utiliza o Samba como Primary Domain Controller (PDC):

- Login no domínio para clientes NT e Windows;
- Perfis de usuário;
- Políticas de segurança do sistema.

A implementação de um PDC Samba pode ser dividida em duas fases:

1. Configurar o PDC Samba;
2. Criar as relações de confiança das máquinas e adicionar os clientes ao domínio.

## Configurando o PDC Samba

O primeiro passo para criar um PDC funcional é entender os parâmetros necessários no *smb.conf*. Todos os parâmetros são encontrados nas *man pages* do Samba. Não detalharemos todos os parâmetros. No entanto, é válido ressaltar alguns detalhes:

- É necessário que as senhas sejam criptografadas (encrypt passwords = yes);
- O servidor precisa suportar logins de domínio;
- O servidor necessita do compartilhamento chamado [netlogon];
- O servidor deve ser o browser mestre do domínio para que os clientes Windows possam identificá-lo como controlador de domínio;

É necessário utilizar o parâmetro global “domain admin group” para definir os usuários e grupos considerados administradores do domínio.

Criação de contas de relação de confiança.

- Conta utilizada para autenticar uma máquina cliente no servidor.
  - ▣ Senha serve para que a máquina se comunique com o controlador de domínio.

Num PDC, cada conta é armazenada em duas partes.

- Uma conta do Samba, armazenada em:
  - ▣ `smbpasswd`
- Uma conta do Linux, armazenada em:
  - ▣ `/etc/passwd`



## Criação das contas de relação de confiança

Uma conta de relação de confiança é uma conta utilizada para autenticar uma máquina cliente (em vez de um usuário) no servidor.

A senha da conta de relação de confiança serve para que a máquina se comunique em segurança com o controlador de domínio, evitando que uma máquina não autorizada com o mesmo nome de NetBIOS se adicione ao domínio e obtenha acesso às contas dos usuários. Os clientes utilizam contas de relação de confiança.

Um PDC Samba armazena cada conta em duas partes:

- Uma conta do Samba, armazenada no arquivo `smbpasswd`;
- Uma conta do Linux, normalmente armazenada em `/etc/passwd`.

Dois modos de criação:

- Manual:
  - ▣ Deve-se criar uma conta Linux e uma conta no Samba.
- Dinâmica:
  - ▣ É necessário configurar:
    - ▣ `smb.conf`
  - ▣ Adicionar o parâmetro:
    - ▣ `add user script`

A adição de clientes ao domínio varia de acordo com a versão do Windows.

Há duas maneiras de criar contas de relação de confiança:

- **Criação manual:** cria normalmente a conta do Samba e a conta Linux;
- **Criação dinâmica:** cria a conta de relação de confiança no momento em que o cliente é adicionado ao domínio.

A criação dinâmica é mais recomendada por motivos de segurança.



## Criação manual de contas de relação de confiança

Primeiro, é necessário criar uma conta Linux em `/etc/passwd`, o que pode ser feito utilizando o comando `useradd`. Por exemplo:

```
# useradd -g 100 -d /dev/null -c "machine nickname" -s /bin/false
machine_name$
# passwd -l machine_name$
```

Essa entrada no arquivo `/etc/passwd` lista o nome da máquina com um caractere \$ adicionado no final; não possui senha e shell. Por exemplo, a máquina chamada tunisia possuiria, nesse caso, as seguintes características:

```
tunisia$:x:505:501:machine nickname:/dev/null:/bin/false
```

Onde *machine nickname* pode ser uma descrição que identifica a máquina cliente e *machine\_name* indica o nome NetBIOS da máquina cliente que será adicionada ao domínio. O caractere \$ deve ser adicionado ao final do nome NetBIOS, ou o Samba não reconhecerá a conta como de relação de confiança.

Uma vez que a conta Linux esteja criada, o próximo passo é criar uma conta no Samba, com o comando `smbpasswd`:

```
# smbpasswd -a -m machine_name
```

## Criação dinâmica de contas de relação de confiança

Essa criação é feita através da configuração do servidor Samba, de modo que ele crie as contas quando necessário, no momento em que máquinas clientes são adicionadas ao domínio. Para tanto, é necessário adicionar o parâmetro “add user script” no arquivo de configuração `smb.conf`.

A seguir é apresentado um exemplo de script para a criação de usuário para um sistema Red Hat:

```
[global]

...

add user script = /usr/sbin/useradd -d /dev/null -g 100 -s /bin/false
-M %u
```

## Adição de clientes no domínio

Esse processo varia de acordo com a versão do Windows. No Windows 2000, por exemplo, quando um usuário faz uma requisição para adicionar uma máquina ao domínio, o Windows pede uma conta e uma senha que tenham privilégios para adicionar clientes no domínio. Uma conta de administrador do Samba deve ser informada.

No Windows NT, por outro lado, se a conta de relação de confiança foi criada manualmente no servidor Samba, adicione o nome do domínio em “Identification Changes”, mas não marque a opção “Create a Computer Account in the Domain”, uma vez que nesse caso a conta já criada é utilizada para adicionar a máquina no domínio. Se a conta de relação de confiança é criada dinamicamente, adicione o nome do domínio no menu “Identification Changes” e selecione “Create a Computer Account in the Domain”.





# Roteiro de Atividades 10

## Atividade 10.1 – Configuração do servidor Samba

Esta atividade deve ser feita pelo cliente Windows. É importante que tanto a máquina cliente quanto o servidor estejam na mesma rede. Como sugestão, vamos considerar que a máquina virtual Windows está com o endereço IP 192.168.1.3 e o servidor Linux, com o endereço IP 192.168.1.1.

Samba é um servidor e um conjunto de ferramentas que permite que máquinas Linux e Windows se comuniquem entre si, compartilhando serviços (arquivos, diretórios e impressão) através dos protocolos Server Message Block (SMB) e Common Internet File System (CIFS), equivalentes à implementação NetBEUI no Windows. O Samba é uma das soluções em ambientes Unix capaz de interligar redes heterogêneas.

### Instalando o Samba

Efetue a instalação do servidor Samba conforme orientação do instrutor:

Durante a instalação poderão ser feitas as seguintes perguntas:

P: Nome do Domínio/Grupo de Trabalho

R: EMPRESA

P: Modificar smb.conf para usar as configurações WINS fornecidas via DHCP?

R: Não

Verificação da versão do Samba:

```
#smbd -V
```

### Configurando o LDAP

#### a. Adicionar o schema do samba à base de dados do LDAP:

Copiar o arquivo /usr/share/doc/samba-doc/examples/LDAP/samba.schema.gz para o diretório de schemas do ldap.

```
# cp /usr/share/doc/samba-doc/examples/LDAP/samba.schema.gz /etc/ldap/schema
```

Descompactar o arquivo

```
#gunzip /etc/ldap/schema/samba.schema.gz
```

Parando o serviço do ldap

```
# /etc/init.d/slaped stop
```





b. Criar o arquivo `/root/ldap_schema.conf` contendo as linhas a seguir:

```
include /etc/ldap/schema/core.schema
include /etc/ldap/schema/collective.schema
include /etc/ldap/schema/corba.schema
include /etc/ldap/schema/cosine.schema
include /etc/ldap/schema/duaconf.schema
include /etc/ldap/schema/dyngroup.schema
include /etc/ldap/schema/inetorgperson.schema
include /etc/ldap/schema/java.schema
include /etc/ldap/schema/misc.schema
include /etc/ldap/schema/nis.schema
include /etc/ldap/schema/openldap.schema
include /etc/ldap/schema/ppolicy.schema
include /etc/ldap/schema/samba.schema
```

c. Criar o diretório `/tmp/output_ldif`

```
# mkdir /tmp/output_ldif
```

d. Utilize o comando `slapcat` para converter os arquivos de schema.

```
# slapcat -f /root/ldap_schema.conf -F /tmp/output_ldif -n0 -s
"cn={12}samba,cn=schema,dc=config" > /root/samba.ldif
```

e. Edite o arquivo `/root/samba.ldif` conforme a seguir:

No topo do arquivo altere:

```
dn: cn={12}samba,cn=schema,cn=config
```

```
...
```

```
cn: {12}samba
```

*Para:*

```
dn: cn=samba,cn=schema,cn=config
```

```
...
```

```
cn: samba
```

Remova as últimas linhas do arquivo:

```
structuralObjectClass: olcSchemaConfig
```

```
entryUUID: 8c5a75ca-963f-187d-acff-2f64fd123c95
```

```
creatorsName: cn=config
```

```
createTimestamp: 20080827045234Z
```

```
entryCSN: 20120523452234.309955Z#000000#000#000000
modifiersName: cn=config
modifyTimestamp: 20120527045345Z
```

f. Adicione o schema à base do LDAP:

```
# ldapadd -Y EXTERNAL -H ldapi:/// -f /root/samba.ldif
```

g. Crie o arquivo chamado `/root/samba_index.ldif` contendo os índices que devem ser criados para o funcionamento do Samba.

```
dn: olcDatabase={1}hdb,cn=config
changetype: modify
add: olcDbIndex
olcDbIndex: uidNumber eq
olcDbIndex: gidNumber eq
olcDbIndex: loginShell eq
olcDbIndex: memberUid eq,pres,sub
olcDbIndex: uniqueMember eq,pres
olcDbIndex: sambaSID eq
olcDbIndex: sambaPrimaryGroupSID eq
olcDbIndex: sambaGroupType eq
olcDbIndex: sambaSIDList eq
olcDbIndex: sambaDomainName eq
olcDbIndex: default sub
```

h. Utilize o comando `ldapmodify` para alterar os índices da base de dados.

```
# ldapadd -Y EXTERNAL -H ldapi:/// -f /root/samba_index.ldif
```

## Configurando o smbldap-tools

```
Obtendo o SID
# net getlocalsid

Anote este numero, pois vamos precisar dele no futuro

Acesso o diretório do smbldap-tools
# cd /etc/smbldap-tools

Copiar os modelos dos arquivos de configuração
#cp /usr/share/doc/smbldap-tools/examples/smbldap.conf.gz .
#cp /usr/share/doc/smbldap-tools/examples/smbldap_bind.conf .

Descompactar o arquivo smbldap.conf
```



```
# gunzip smbldap.conf.gz
Configuração smbldap.conf
SID="S-1-5-21-1669739004-1474045143-3185410112"
sambaDomain="EMPRESA"
slaveLDAP="192.168.0.1"
slavePort="389"
masterLDAP="192.168.0.1"
masterPort="389"
ldapTLS="0"
ldapSSL="0"
verify="none"
suffix="dc=empresa,dc=com,dc=br"
usersdn="ou=People,${suffix}"
computersdn="ou=Hosts,${suffix}"
groupsdn="ou=Group,${suffix}"
idmapdn="ou=Idmap,${suffix}"
sambaUnixIdPooldn="sambaDomainName=${sambaDomain},${suffix}"
scope="sub"
hash_encrypt="SSHA"
crypt_salt_format="%s"
userLoginShell="/bin/bash"
userHome="/home/%U"
userHomeDirectoryMode="700"
userGecos="System User"
defaultUserGid="513"
defaultComputerGid="515"
skeletonDir="/etc/skel"
defaultMaxPasswordAge="45"
userSmbHome="//SERVIDOR/%U"
userProfile="//SERVIDOR/%U\\profile"
userHomeDrive="H:"
userScript="netlogon.bat"
mailDomain="empresa.com.br"
```

```

with_smbpasswd="0"
smbpasswd="/usr/bin/smbpasswd"
with_slappasswd="0"
slappasswd="/usr/sbin/slappasswd"
Configuração smbldap_bind.conf
slaveDN="cn=admin,dc=empresa,dc=com,dc=br"
slavePw="rnpesr"
masterDN="cn=admin,dc=empresa,dc=com,dc=br"
masterPw="rnpesr"
Popule o banco do ldap com as informações novas
# smbldap-populate

```

## Configurando o Samba

```

[global]

workgroup = EMPRESA

netbios name = servidor

server string = %h server

dns proxy = no


log level = 2

log file = /var/log/samba/log.%m

max log size = 1000

syslog = 0

panic action = /usr/share/samba/panic-action %d


security = user

admin users = root

encrypt passwords = true

passdb backend = ldapsam:ldap://192.168.1.1/

obey pam restrictions = no

    unix password sync = no

ldap passwd sync = yes

passwd program = /usr/sbin/smbldap-passwd %u

passwd chat = *Enter\snew\s*\spassword:* %n\n *Retype\snew\s\

```

```

spassword:* %n\n *password\supdated\ssuccessfully* .
pam password change = yes

ldap ssl = no
ldap admin dn = cn=admin,dc=empresa,dc=com,dc=br
ldap suffix = dc=empresa,dc=com,dc=br
ldap group suffix = ou=Group
ldap user suffix = ou=People
ldap machine suffix = ou=Hosts
ldap idmap suffix = ou=Idmap
ldap delete dn = Yes
add user script = /usr/sbin/smbldap-useradd -m "%u"
add machine script = /usr/sbin/smbldap-useradd -t 0 -w "%u"
add group script = /usr/sbin/smbldap-groupadd -p "%g"
add user to group script = /usr/sbin/smbldap-groupmod -m "%u" "%g"
delete user script = /usr/sbin/smbldap-userdel "%u"
delete group script = /usr/sbin/smbldap-groupdel "%g"
delete user from group script = /usr/sbin/smbldap-groupmod -x "%u"
"%g"
set primary group script = /usr/sbin/smbldap-usermod -g "%g" "%u"
domain master = yes
domain logons = yes
enable privileges = yes
logon path = \\%N%\%U\profile
logon home = \\%N%\%U
logon script = netlogon.bat
[homes]
comment = Home Directories
browseable = no
read only = no
create mask = 0700
directory mask = 0700
valid users = %S

```

```

[netlogon]

    comment = Network Logon Service
    path = /home/samba/netlogon
    guest ok = yes
    browseable = no

[profiles]

    comment = Users profiles
    path = /home/samba/profiles
    guest ok = no
    browseable = no
    create mask = 0600
    directory mask = 0700

[printers]

    comment = All Printers
    browseable = no
    path = /var/spool/samba
    printable = yes
    guest ok = no
    read only = yes
    create mask = 0700

[print$]

    comment = Printer Drivers
    path = /var/lib/samba/printers
    browseable = yes
    read only = yes
    guest ok = no

Verificar as configurações com o comando testparm

# testparm

Criar o diretorio /home/samba/netlogon

# mkdir -p /home/samba/netlogon

Criar o diretorio /home/samba/profiles

# mkdir -p /home/samba/profiles

# chmod 777 /home/samba/profiles

```

```
Armazenar a senha do LDAP
```

```
# smbpasswd -w rnpesr
```

```
Reiniciar o serviço do samba
```

```
# /etc/init.d/samba restart
```

```
Criar o usuário root do samba
```

```
# smbpasswd -a root
```

## Atividade 10.2 – Testando o funcionamento do Samba

---

1. Adicione o usuário “alunosmb” no LDAP;
2. Adicione a estação Windows utilizando os scripts do smbldap-tools.

## Atividade 10.3 – Windows no domínio do Samba

---

Para esta atividade o aluno deverá iniciar sua máquina virtual Windows e certificar-se de que ela está com o endereço IP 192.168.1.3.

1. Faça o login como administrador;
2. Clique em “Iniciar” > “Painel de Controle” > “Sistema e Segurança” > “Sistema” > “Configurações Avançadas de Sistema”.
3. Na janela que se abrirá, clique sobre a aba “Nome do Computador”.
4. Clique sobre o botão “Alterar”.
5. Marque a opção “Membro de Domínio”.
6. Informe o nome do domínio do samba: “EMPRESA”.
7. Clique sobre o botão “OK” e aguarde.
8. Entre com o login e senha do usuário root do Samba.
9. Clique em “OK” e aguarde a mensagem de boas-vindas.
10. Reinicie o Windows e faça o login no domínio EMPRESA com o usuário alunosmb.

## Atividade 10.4 – Compartilhamentos

---

1. Crie um o diretório `/home/dados` e utilize o Samba para compartilhá-lo com os usuários cadastrados em seu sistema. O nome do compartilhamento deverá ser `[dados]`;
2. Cadastre um novo usuário no samba chamado “alunosmb2” e realize o login na estação Windows;
3. Com o usuário alunosmb2m, acesse o compartilhamento `\\servidor\dados`, grave um arquivo com a data atual e faça o logout do Windows;
4. Acesse a máquina Windows com o usuário alunosmb e verifique se o arquivo criado pelo usuário alunosmb2 está disponível no compartilhamento;
5. Crie um script para mapear automaticamente o compartilhamento `[dados]` na unidade M: sempre que um usuário efetuar o logon no domínio.

## Atividade 10.5 – Excluindo usuários

---

Remova os usuários `alunosmb` e `alunosmb2`, criados nas atividades anteriores. Não se esqueça de verificar se foram removidos tanto do LDAP quando do Samba.

## Atividade 10.6 – Conhecendo o SWAT

---

SWAT é uma interface web para configuração. Uma grande vantagem do SWAT é permitir a configuração remota de um servidor Samba, já que só é necessário acessar a máquina pela internet.

1. Verifique se o SWAT está instalado no servidor.
2. Para habilitar o SWAT, crie o arquivo `/etc/xinetd.d/swat` conforme abaixo:

```
service swat {  
    port = 901  
    socket_type = stream  
    wait = no  
    only_from = 192.168.1.1  
    user = root  
    server = /usr/sbin/swat  
    log_on_failure += USERID  
    disable = no  
}
```

3. No servidor Samba, abra o navegador e acesse o endereço: <http://localhost:901>





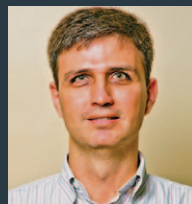
# Bibliografia

- ARKILLS, Brian. *LDAP Directories Explained – An Introduction and Analysis*. Addison-Wesley, 2003.
- CHIN, Liou Kuo. *Rede Privada Virtual – VPN. News Generation*. Boletim bimestral sobre tecnologia de redes. Rede Nacional de Ensino e Pesquisa (RNP). Volume 2, número 8, 1998.
- Documentação de Domínios Virtuais Apache: <http://httpd.apache.org/docs/2.2/vhosts/>
- Documentação de Módulos Apache: <http://modules.apache.org>
- Documentação Oficial Apache: <http://www.apache.org/docs>
- FERREIRA, Rubem E. *Linux – Guia do Administrador do Sistema*. Rio de Janeiro: Editora Novatec, 2003.
- FILHO, João Eriberto Mota. *Descobrindo o Linux*. 2ª ed. Rio de Janeiro: Editora Novatec, 2007.
- HILDEBRANDT, Ralf. *Postfix Start of the Art Message Transport*. No Starch Press, 2005.
- KUROSE, James F.; ROSS, Keith W. *Redes de Computadores e a Internet*. Pearson Addison-Wesley, 2006.
- LANGFELDT, N. DNS Howto: <http://www.tldp.org/HOWTO/DNS-HOWTO.html>
- LAUREANO, Marcos. *Máquinas virtuais e emuladores: conceitos, técnicas e aplicações*. São Paulo: Novatec, 2006.
- LIMA, João P. *Administração de Redes Linux*. Editora Terra, 2003.
- Linux Magazine: [http://www.linux-magazine.com/w3/issue/86/Kernel\\_Based\\_Virtualization\\_With\\_KVM.pdf](http://www.linux-magazine.com/w3/issue/86/Kernel_Based_Virtualization_With_KVM.pdf)
- LIU, C.; ALBITZ, P. *DNS & BIND*. 5ª ed. O' Reilly Media, 2006.
- MARCELO, Antônio. *Squid – Configurando o Proxy para Linux*. Brasport, 2006.
- NFS Linux: [nfs.sourceforge.net/nfs-howto](http://nfs.sourceforge.net/nfs-howto)
- NFSv4: [www.nfsv4.org](http://www.nfsv4.org)
- OpenLDAP: <http://www.openldap.org>
- Openswan: <http://www.openswan.org>

- ▣ PINTO, L.; CESARIO M.; Monteiro. M. *Cache: melhor aproveitamento dos recursos na Internet*. Boletim bimestral sobre tecnologia de redes da Rede Nacional de Ensino e Pesquisa (RNP). Volume 1, número 2, 1997.
- ▣ RFC 1034 – Domain names: concepts and facilities
- ▣ RFC 1035 – Domain names: implementation and specification
- ▣ RFC 1661 – The Point-to-Point Protocol (PPP)
- ▣ RFC 1939 – Post Office Protocol (POP3) – Version 3
- ▣ RFC 2131 – Dynamic Host Configuration Protocol
- ▣ RFC 2132 – DHCP Options and BOOTP Vendor Extensions
- ▣ RFC 2616 – Hypertext Transfer Protocol: HTTP/1.1
- ▣ RFC 3396 – Encoding Long Options in the Dynamic Host Configuration Protocol (DHCPv4)
- ▣ RFC 3501 – IMAPv4 – Internet Message Access Protocol – Version 4rev1
- ▣ RFC 3947 – Negotiation of NAT-Traversal in the IKE
- ▣ RFC 4251 – The Secure Shell (SSH) Protocol Architecture
- ▣ RFC 4510 – Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map
- ▣ RFC 4511 – Lightweight Directory Access Protocol (LDAP): The Protocol
- ▣ RFC 4519 – Lightweight Directory Access Protocol (LDAP): Schema for User Applications
- ▣ RFC 765 – File Transfer Protocol (FTP)
- ▣ RFC 821 – Simple Mail Transfer Protocol (SMTP)
- ▣ RFC 854 – Telnet Protocol Specification
- ▣ RFC 855 – Telnet Option Specifications
- ▣ RICCI, Bruno. *Squid – Solução definitiva*. Rio de Janeiro: Ciência Moderna, 2005.
- ▣ Samba Project: <http://www.samba.org>
- ▣ Squid: <http://www.squid-cache.org>
- ▣ Squirm: <http://squirm.foote.com.au>
- ▣ TANENBAUM, Andrew S.; STEEN, Maarten Van. *Sistemas Distribuídos: princípios e paradigmas*. 2º edição. Editora Pearson, 2010.
- ▣ TERPSTRA, J. *Samba-3 by Example*: <http://us4.samba.org/samba/docs>
- ▣ The IMAP Connection: <http://www.imap.org>
- ▣ The Postfix Homepage: <http://www.postfix.org>
- ▣ TS, Jay; ECKSTEIN, Robert; COLLIER-BROWN, David. *Using Samba*. 2ª ed.
- ▣ VERNOOIJ, R.; TERPSTRA, J.; CARTER, G. *The Official Samba-3 Howto and Reference Guide*. <http://us4.samba.org/samba/docs>
- ▣ <http://libvirt.org>

- ▣ [http://pic.dhe.ibm.com/infocenter/lnxinfo/v3r0m0/topic/laat/liaatbestpractices\\_pdf.pdf](http://pic.dhe.ibm.com/infocenter/lnxinfo/v3r0m0/topic/laat/liaatbestpractices_pdf.pdf)
- ▣ <http://pubs.vmware.com/vsphere-51/index.jsp>
- ▣ <http://technet.microsoft.com/en-us/windowsserver/dd448604>
- ▣ <http://us4.samba.org/samba/docs>
- ▣ <http://wiki.qemu.org/Manual>
- ▣ <http://www.gta.ufrj.br/ensino/CPE758/artigos-basicos/cap4-v2.pdf>
- ▣ [http://www.gta.ufrj.br/grad/08\\_1/virtual/artigo.pdf](http://www.gta.ufrj.br/grad/08_1/virtual/artigo.pdf)
- ▣ [http://www.juliobattisti.com.br/artigos/windows/tcpip\\_p9.asp](http://www.juliobattisti.com.br/artigos/windows/tcpip_p9.asp)
- ▣ <http://www.linux-kvm.org/>
- ▣ <http://www.rootlinux.com.br/documentos/downloads/apostila-dns.pdf>
- ▣ <http://www.tldp.org/HOWTO/DHCP/index.html>
- ▣ <http://www.tldp.org/HOWTO/DNS-HOWTO.html>
- ▣ <http://www.xen.org/support/documentation.html>
- ▣ [https://access.redhat.com/site/documentation/pt-BR/Red\\_Hat\\_Network\\_Satellite/5.4/html/Reference\\_Guide/ch-virtualization.html](https://access.redhat.com/site/documentation/pt-BR/Red_Hat_Network_Satellite/5.4/html/Reference_Guide/ch-virtualization.html)





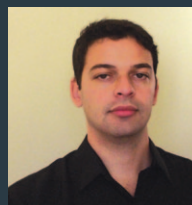
**Eduardo Lobo** Analista de Informática no Ministério Público do Trabalho (MPT) e Professor dos Cursos de TI da Universidade Católica de Brasília. Engenheiro Mecânico graduado pela Universidade Federal de Uberlândia e Mestre em

Ciência da Computação pela Universidade Federal de Santa Catarina, Florianópolis, 2001. Atuou em grandes empresas no período de 1997-2000, como Analista de TI para o governo brasileiro em 2001, e como Coordenador dos Cursos de Pós-Graduação Lato Sensu em Redes de Computadores da Universidade Católica de 2003 a 2005, e de Tecnologia em Segurança da Informação (TSI) de 2007 a 2009. Participou do Projeto de Pesquisa CESMIC/UCB de 2003 a 2008. Atualmente administra o datacenter e a rede de dados/voz do MPT. Áreas de interesse incluem: redes de computadores, storage e sistemas baseados em software livre.



**Wagner Vieira Léo** tem 25 anos de experiência na área de TI, atuando como Analista de Suporte e em Computação de Alto Desempenho, com foco em sistemas operacionais Unix/Linux. Possui graduação em Matemática pela Faculdade de Humanidades Pedro II, com Pós-Gradua-

ção em Gestão da Inovação pelo LNCC/UCP. Atualmente ocupa o cargo de Coordenador de Tecnologia da Informação do Laboratório Nacional de Computação Científica e de Coordenador Administrativo do Ponto de Presença da RNP no Rio de Janeiro. Professor do Instituto Superior de Tecnologia da Informação de Petrópolis, IST e da Escola Superior de Redes da RNP, nas áreas de Segurança da Informação e Sistemas Operacionais.



**Bruno Alves Fagundes** é tecnólogo em Tecnologia da Informação e da Computação pelo Instituto Superior de Tecnologia (2007) e especialista em Segurança de Redes pela Universidade Estácio de Sá (2011). Tem experiência em administração de servidores Linux,

gerenciamento de usuário, implementação e gerência de serviços de rede, clusters, segurança de redes, shell script, PHP, Perl. Atualmente é colaborador do LNCC, onde trabalha na administração dos servidores e clusters do Laboratório de Bioinformática (LABINFO), provendo infraestrutura para a realização de pesquisas e processamento massivo de dados.



**Francisco Marcelo M. Lima** é certificado Project Management Professional (PMP) e Modulo Certified Security Officer (MCSO), Mestre em Engenharia Elétrica pela Universidade de Brasília (2009), Mestre em Liderança pela Universidade de Santo Amaro (2007) e pós-graduado em Segurança de Redes de

Computadores pela Universidade Católica de Brasília (2003). Atualmente exerce as funções de Coordenador dos Cursos de Redes de Computadores e Segurança da Informação do IESB, e de Analista em TI do MPOG cedido para a Controladoria-Geral da União/PR. Possui mais de 15 anos de experiência na área de Ciência da Computação, com ênfase em Segurança da Informação, Redes e Construção de Software.

O curso ensina a projetar, instalar, configurar e disponibilizar os principais serviços para internet em uma rede TCP/IP. Apresenta os conceitos associados a cada um dos serviços, e a instalação e configuração do KVM como base para o ambiente de virtualização. Aborda a autenticação nos serviços com LDAP, com apoio intensivo de atividades práticas.

Este livro inclui os roteiros das atividades práticas e o conteúdo dos slides apresentados em sala de aula, apoiando profissionais na disseminação deste conhecimento em suas organizações ou localidades de origem.

ISBN 978-85-63630-22-3



9 788563 630223